Universidad Pública de Navarra

ESCUELA TECNICA SUPERIOR DE INGENIEROS AGRONOMOS

Nafarroako Unibertsitate Publikoa

NEKAZARITZAKO INGENIARIEN GOI MAILAKO ESKOLA TEKNIKOA

DETECCIÓN AUTOMÁTICA DE LINEAS ELÉCTRICAS DE ALTA TENSIÓN EN DATOS LIDAR MEDIANTE MINERÍA DE DATOS

presentado por

DANIEL CHASCO HERNÁNDEZ

aurkeztua

MÁSTER EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA Y TELEDETECCIÓN
MASTERRA INFORMAZIO SISTEMA GEOGRAFIKOETAN ETA TELEDETEKZIOAN

SEPTIEMBRE 2018 / *2018, IRAILA*

Abstract

The correct classification of power lines has been one of the most important topics of mapping. The aim of this research is the detection and automatic extraction of high-voltage transmission lines from LIDAR data using data mining. The LIDAR dataset has been obtained using SPL (Single Photon LiDAR) technology during the mapping project of the LIDAR data capture in the Autonomous Community of Navarre realized by the company Tracasa in 2017.

Keywords: LIDAR, data mining, Single Photon Lidar, power lines, supervised classification

Agradecimientos

A la empresa TRACASA donde realice mis practicas por proporcionarme todo lo necesario para la realización del trabajo y en especial a Víctor por aguantar las dificultades todo ese tiempo.

Al Jesús y Josean director y codirector del presente trabajo por las correcciones y los ánimos dados, necesarios para poder llevarlo a buen fin.

Índice de contenido.

Índice de figuras

Índice de tablas

1- PRESENTACIÓN

1.1 Introducción

De todas las infraestructuras de distribución energética de un país las redes de distribución de electricidad sean quizás la parte más importante. La correcta clasificación de estas líneas eléctricas en la cartografía se presenta como una tarea costosa en cuanto a tiempo de trabajo para los organismos encargados de realizarla (Zhu & Hyyppä, 2014). Con este trabajo se busca la creación de un algoritmo basado en minería de datos que sea capaz de detectar automáticamente las líneas eléctricas de alta tensión y clasificarlas correctamente a partir de los datos LIDAR obtenidos durante 2017 en el proyecto LiDAR de Navarra mediante la tecnología *Single Photon Lidar*.

El sistema LIDAR se ha introducido en el ámbito de la cartografía como un método barato y eficaz para la recogida de datos que es capaz de capturar rápidamente nubes de puntos en 3D (Marsh, 2016). Además, permite una automatización en la recogida de datos y la generación de mapas con un nivel de detalle hasta ahora inalcanzable con otras técnicas cartográficas (Y Jwa, Sohn, & Kim, 2009). Antes de la llegada de los actuales sistemas LIDAR tanto el levantamiento topográfico como la creación de mapas a partir de técnicas fotogramétricas requerían una gran cantidad de mano de obra, por lo que resultaba costoso trazar mapas con un alto nivel de detalle (Dawe & Engineer, 1947). Desde la llegada de los sensores LIDAR esta tarea puede realizarse de forma más rápida y económica.

Los datos LIDAR pueden producir mapas topográficos muy detallados con gran precisión. Estos datos pueden ser utilizados en muchas aplicaciones como extracción de modelos digitales del terreno (MDT), modelos digitales de elevaciones (MDE), labores de dasometría e inventario forestal, riesgo de avenidas, aplicación de fitosanitarios, clasificación de usos de suelo, labores arqueológicas, etc (Marsh, 2016). En este trabajo de investigación son utilizados para la detección, mediante técnicas de minería de datos, de las líneas eléctricas de alta tensión presentes en la zona de estudio.

1.2 Objetivos

El objetivo principal de este trabajo será la detección de cables pertenecientes a líneas eléctricas de alta tensión a partir de datos LIDAR mediante algoritmos inteligentes de extracción de información.

Para conseguir este objetivo se hace uso del conjunto de datos LIDAR obtenido mediante la tecnología SPL (*Single Photon LiDAR*) desarrollada por *Leica geosystem* en el ámbito del proyecto cartográfico de captura de datos LIDAR en la Comunidad Foral de Navarra realizado por la empresa navarra *Tracasa* durante el año 2017.

La consecución del objetivo principal lleva asociados los siguientes objetivos específicos:

- Desarrollar una metodología lo más automática posible y que pueda ser extrapolable y aplicable fácilmente a otros entornos. Esta metodología estará basada en la aplicación de algoritmos de clasificación supervisada en los datos LIDAR de Navarra del proyecto citado anteriormente.

- Evaluar la detección de cables en función de las áreas de entrenamiento, permitiendo concluir si realmente el uso de esta información constituye un elemento clave en la detección posterior.
-

2- ANTECEDENTES

2.1 Tecnología LIDAR

2.1.1 Introducción

El LiDAR (Light Detection And Ranging) es un sistema activo de detección remota basado en un escáner láser (Landa et al., 2013). Este sistema obtiene una muestra de puntos de la superficie de la tierra produciendo mediciones de las coordenadas de X, Y y Z (Mozo & Alconada, 2014). Los datos obtenidos con el sistema LiDAR se utilizan principalmente en aplicaciones de representación cartográfica. Debido a sus características se está convirtiendo en una alternativa rentable para las técnicas de cartografía y topografía tradicionales como la fotogrametría (Fagua, Campo, & Posada, 2011).

Las partes que componen un sistema LiDAR como podemos ver en la figura 1 incluyen generalmente un GPS para medir la posición exacta y un INS (sistema de navegación por inercia) o un GNSS/IMU – (Unidad de Medidas Inerciales). (Conama, 2016). Estos sensores se integran dentro del vehículo aéreo o terrestre para tener información acerca del movimiento del propio vehículo y de su localización.

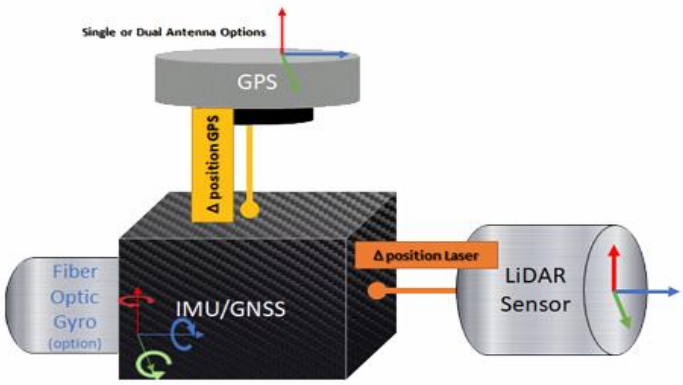


Figura 1: Partes de un sistema LIDAR. Fuente: www.modusrobotics.com/lidarsystems/

El funcionamiento de un sistema LIDAR desde una aeronave como podemos ver en la figura 2 está basado en la emisión de rayos láser hacia un objetivo desde una plataforma o vehículo aéreo. La medición precisa del tiempo de retorno de las porciones del pulso al sensor permite calcular la distancia que separa a éste de la superficie terrestre y de los objetos que existen sobre ella. La información recibida de esta medición combinada con la información de la posición (GPS e INS) hace que sea posible la transformación de estas medidas en puntos tridimensionales reales del objetivo (Romero et al., 2009).

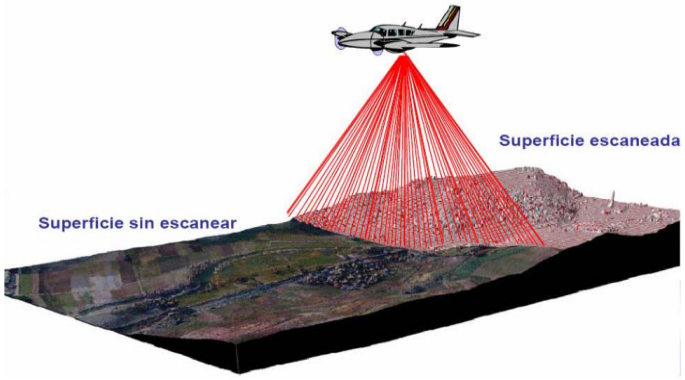


Figura 2: Esquema de trabajo de un sistema LiDAR. Fuente: (Albacete, 2011)

Estos puntos se procesan posteriormente y se les asignan los atributos de coordenadas X, Y, Z georreferenciadas, el rango de tiempo que tarda el punto en volver al sensor, el ángulo de escaneo láser, la posición del GPS y la información del INS (ArcGis Resources, 2016).

La gran mayoría de objetos presentes en el terreno reflejan los pulsos laser emitidos desde el sensor y vuelven al propio sensor. Si el terreno no presente ningún elemento opaco a los rayos laser, como un edificio, generalmente también llegan pulsos procedentes de la superficie del terreno. Por lo tanto, podemos tener reflejos en vegetación, edificios, y cualquier elemento del terreno. Se podría decir que un pulso láser emitido que encuentre varias superficies que lo reflejen en su viaje hacia el suelo se dividirá en tantos retornos como superficies se encuentre antes de llegar al suelo (ArcGis Resources, 2016). Debido a lo anterior un pulso láser puede regresar al sensor como uno o muchos retornos.

En cuanto a la importancia de los retornos de cada pulso laser podemos decir que el primer pulso láser que retorne al sensor generalmente será el más importante y corresponderá a la entidad más grande del terreno como una copa de árbol o la parte superior de un edificio. En casos en los que el haz láser no encuentre nada en su camino hacia el suelo el primer retorno corresponderá al suelo, en cuyo caso el sistema LiDAR solo detectará un retorno (Mozo & Alconada, 2014).

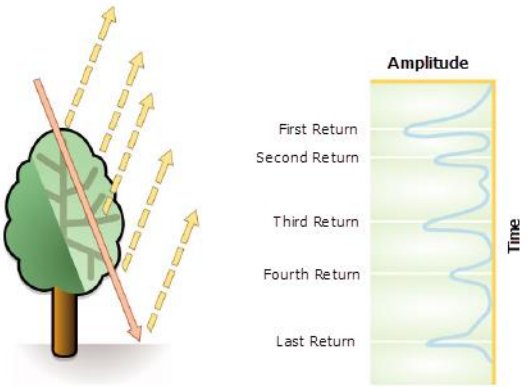


Figura 3: Esquema de los retornos de un pulso LiDAR. Fuente ArcGis Resources, 2016

Se puede concluir diciendo que si nos encontramos con superficies sin vegetación se obtiene un único retorno que corresponderá al suelo, en cambio en superficies con vegetación el sensor detectara varios retornos para un mismo pulso láser ya que éste es reflejado parcialmente por la vegetación. Esta capacidad de poseer múltiples retornos un mismo pulso laser es fundamental para entender las aplicaciones del LiDAR. Gracias a esto una nube de puntos LIDAR permite describir con precisión la estructura de la vegetación capturando información tridimensional de los diferentes estratos y del suelo (Landa et al., 2013).

Además, la capacidad del sensor LIDAR para capturar información bajo la cubierta vegetal hace que podamos obtener información del terreno presente bajo ella. Esta cualidad es una de sus principales ventajas de esta tecnología, ya que el resto de sensores utilizados en teledetección son incapaces de traspasar la vegetación (Albacete, 2011). Generalmente y debido a la alta densidad de puntos generada por el sensor LiDAR, incluso si solo un pequeño porcentaje de puntos alcanza el suelo suele ser suficiente para crear un Modelo Digital del Terreno (Landa et al., 2013).

2.1.2 Estándar de datos: Formato LAS

El formato LAS se ha convertido en el estándar de datos LiDAR (ASPRS, 2013). Este formato se caracteriza por ser un archivo binario que guarda la información procedente de los puntos obtenidos durante el vuelo. En sus inicios se creó como una alternativa a los ficheros propietarios generados por las distintas compañías para facilitar el intercambio de datos LIDAR entre empresas y software de procesamiento (Martínez Blanco, 2016). Actualmente el formato LAS se ha convertido en el formato más utilizada para el intercambio de nubes de datos LIDAR (ASPRS, 2013).

La organización encargada de la creación, distribución y actualización de este formato es la ASPRS (*American Society for Photogrammetry & Remote Sensing)*. El formato está dividido en dos partes: una cabecera constituida por una línea para guardar el número de puntos y los valores de las variables donde se almacena la información de la proyección, metadatos y otros datos de aplicación del usuario y el almacenamiento de los datos a modo de puntos. (Martínez Blanco, 2016)

La ASPRS ha ido modificando las versiones del formato desde su creación en 2003 incluyendo distintas configuraciones en cada formato. La Especificación LAS 1.4 fue aprobada por la Junta Directiva de ASPRS el 14 de noviembre de 2011 y es la versión más reciente aprobada del documento (ASPRS, 2013). En la tabla 1 se puede ver su evolución.

Versión	**Fechas**
LAS 1.0	Mayo de 2003
LAS 1.1	Mayo de 2005
LAS 1.2	Septiembre de 2008
LAS 1.3	Octubre de 2010
LAS 1.4	Noviembre de 2011

Tabla 1: Versiones formato LAS. Fuente ASPRS

Las novedades que presenta la última versión disponible son las siguientes: (ASPRS, 2013).

- Extensión de tamaño de cada campo para soportar 64 bits completos.
- Soporte para hasta 15 retornos por pulso.
- Extensión del campo Point_Class para soportar 256 clases.
- Definición de varias nuevas clases estándar ASPRS.
- Ampliación del campo de ángulo de exploración a 2 bytes para permitir una resolución angular más fina.
- Adición de un bit de solapamiento para permitir la indicación de pulsos en la región de solapamiento.
- Adición de un registro de longitud variable de bytes adicionales (opcional) para describir atributos adicionales almacenados con cada punto.

En el último formato (1.4), el cual va a ser el utilizado en el presente trabajo, para cada punto LiDAR disponemos de la información que se muestra en la tabla 2.

CAMPO	**DESCRIPCIÓN**
X	Coordenada X
Y	Coordenada Y
Z	Coordenada Z
Intensity	Intensidad del punto láser a la llegada al sensor.
Return_Number	Numero de retorno de ese pulso.
Number_of_Returns	Numero de retornos detectados en ese pulso.
R	Valor asociado al canal Rojo.
G	Valor asociado al canal Verde.
B	Valor asociado al canal Azul.
Classification	Clasificación asignada a ese punto.
Scan_Direction_flag	Dirección del espejo del escáner.
Edge_of_flight_line	Borde de línea de vuelo.
Scan_Angle	Angulo de escaneo.
User_Data	Campo a rellenar por el usuario según sus necesidades.
Point_Sourde_ID	Identificador de pasada.

Tabla 2: Campos de cada punto LiDAR en un archivo LAS. Fuente ASPRS

A continuación describimos el significado de los campos almacenados en los archivos LAS.

- ***X, Y, y Z***: Los valores X, Y, y Z se almacenan como enteros y se utiliza junto con los valores de la escala y los valores de desviación para determinar la coordenada de cada punto (Soininen, 2016).

- ***Intensity*****:** Es la representación entera de la magnitud de retorno del pulso. Este es opcional y específico del sistema. Sin embargo, siempre debe incluirse si está disponible. La intensidad, cuando se incluye, siempre se normaliza a un valor de 16 bits, sin signo, multiplicando el valor por 65,536 (rango dinámico de intensidad del sensor). Por ejemplo, si el rango dinámico del parámetro es de 10 bits, el valor de escala sería (65,536/1,024). Si no se incluye la intensidad, este valor debe ponerse a cero. Esta normalización es necesaria para garantizar que los datos de diferentes sensores puedan fusionarse correctamente. (ASPRS, 2013)

- ***Return_Number***: El número de retorno es el número de retorno de pulso para un pulso de salida dado. Un determinado pulso de salida puede tener muchos retornos, y deben ser marcados en secuencia de retornos. ASPRS

- ***Number_of_Returns*** (de cada pulso): El número de retornos es el número total de devoluciones de un pulso dado. ASPRS

- ***Scan_Direction_flag:*** Indica la dirección de escaneo del espejo en el momento del pulso de salida. Un valor 1 indica que la dirección del escáner es positiva (el espejo se mueve desde el lado izquierdo, en la dirección del trayecto, al lado derecho), mientras que el 0 significa un movimiento del espejo en sentido contrario (desde el lado derecho al izquierdo). (Martínez Blanco, 2016)

- ***Edge_of_Flight_Line:*** indica el borde de la línea de vuelo. Un valor de 1 cuando se corresponde con puntos del final del escaneo, lo que significa que se va a producir un cambio en la dirección de la pasada. (Martínez Blanco, 2016)

- ***Point_Source_ID***: Este campo está constituido por un valor numérico para marcar la dirección de la pasada a la que pertenece ese punto. La información de este campo debe coincidir con el de *File_Source_ID*. En cuanto al campo *Scan_Angle* señalar que puede adquirir valores entre -90º y +90º, siendo el valor de 0º para los puntos situados en el nadir y -90º para los del lado derecho del avión en la dirección de vuelo.(Martínez Blanco, 2016)

- ***R, G, B***: Son los campos correspondientes a los colores de cada punto RGB (rojo, verde y azul) generalmente vienen de las imágenes recopiladas al mismo tiempo que la captura de puntos LIDAR. Si esto no es posible también se pueden tomar de una orto fotografía de la zona de captura de puntos.

- ***Classification****:* Este valor indica el tipo de elemento de la clasificación. En la tabla 3 podemos observar los valores de clasificación utilizados en el formato LAS en la última versión (1.4). En esta versión se añadieron nuevos campos de clasificación hasta contar con 255 ya que la versión 1.3 solo tenía 31 valores posibles de clasificación.(Martínez Blanco, 2016)

NÚMERO	CLASE	TIPO DE ELEMENTO
0	*Created, never classified*	Nunca clasificado
1	*Unclassified*	No asignado
2	*Ground*	Suelo
3	*Low Vegetation*	Vegetación baja
4	*Medium Vegetation*	Vegetación media
5	*High vegetation*	Vegetación alta
6	*Building*	Edificio
7	*Low Point (noise)*	Punto bajo (ruido)
8	*Reserved*	Reservado
9	*Water*	Agua
10	*Rail*	Vía férrea
11	*Road Surface*	Carretera
12	*Reserved*	Reservado
13	*Wire-Guard (Shield)*	Tendido
14	*Wire-Conductor (Phase)*	Cable eléctrico
15	*Transmision Tower*	Torre eléctrica
16	*Wire-structure Connector*	Conector de tendido
17	*Bridge Deck*	Cubierta de puente
18	*High Noise*	Punto alto (ruido)
19-63	*Reserved*	Reservado
14-255	*User definable*	Definido por el usuario

Tabla 3: Clases existentes en un archivo LAS. Fuente: ASPRS

2.1.3 Tecnología Single Photon LiDAR (SPL)

En este trabajo se utilizan datos procedentes del sensor SPL100 de *Leica Geosystems* el cual utiliza la tecnología LIDAR SPL.

La principal novedad del sistema SPL y que la hace distinta al resto de sensores LIDAR utilizados hasta ahora es que contiene un divisor del haz láser, dividiendo cada pulso láser en un conjunto de diez por diez pequeños rayos láser (Figura 4) (Sirota et al, n.d). Una vez que se han dividido en estos 100 puntos cada uno de ellos actúa como un fotón de un sistema LIDAR tradicional, el tiempo de viaje de los fotones al suelo y de vuelta se mide individualmente. Al dividir cada haz láser en 100 haces diferentes la cantidad de energía de cada punto es mucho menor en comparación con el LiDAR tradicional, pero, gracias a la mayor sensibilidad de estos sensores, un solo fotón de retorno es suficiente para medir un rango (Wulder et al., 2008).

El sistema SPL puede generar 60.000 pulsos por segundo, como cada pulso se divide en 100 partes, esto resulta en una frecuencia de pulso efectiva de 6.0MHz - significativamente más alta que la que se puede lograr con el LiDAR tradicional. (Swatantran, Tang, Barrett, Decola, & Dubayah, 2016)

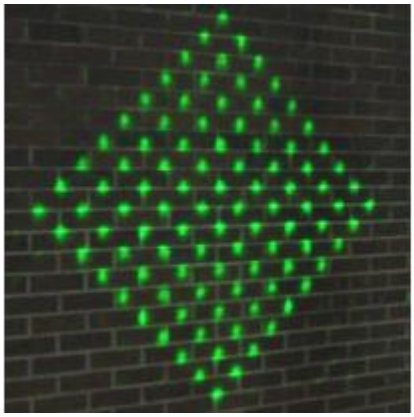


Figura 4: División de un haz láser en 10x10 haces. Fuente: Sirota et al

Los instrumentos LiDAR tradicionales multifotónicos (MPL) de uso común se basan en la digitalización de la forma de la onda o en la captura de los puntos de retorno discreto. En los instrumentos LiDAR que digitalizan la forma de onda, toda la señal de retorno se digitaliza para proporcionar una forma que describa el perfil del terreno. En la MPL de captura de los puntos de retorno discreto, se registran uno o más retornos por cada pulso para obtener una nube de puntos tridimensional que representa la elevación del terreno, el dosel y otros elementos tales como edificios. (Swatantran et al., 2016)

Los sensores SPL no capturan una onda continua como los sistemas multifotónicos, sino que cuentan los fotones individuales, debido a esta característica estos sistemas no pueden recuperar la forma de onda completa como en los sistemas LIDAR tradicionales o multifotónicos (Sirota et al, n.d,). Para compensarlo, estos sistemas recuperan múltiples retornos gracias a los tiempos de recuperación de canal muy cortos de 1,6 nanosegundos. Esto es posible debido a que el contador de fotones se reinicia cada 1,6 nanosegundos para contar si algún fotón nuevo regresa del objetivo. (Wulder et al., 2008)

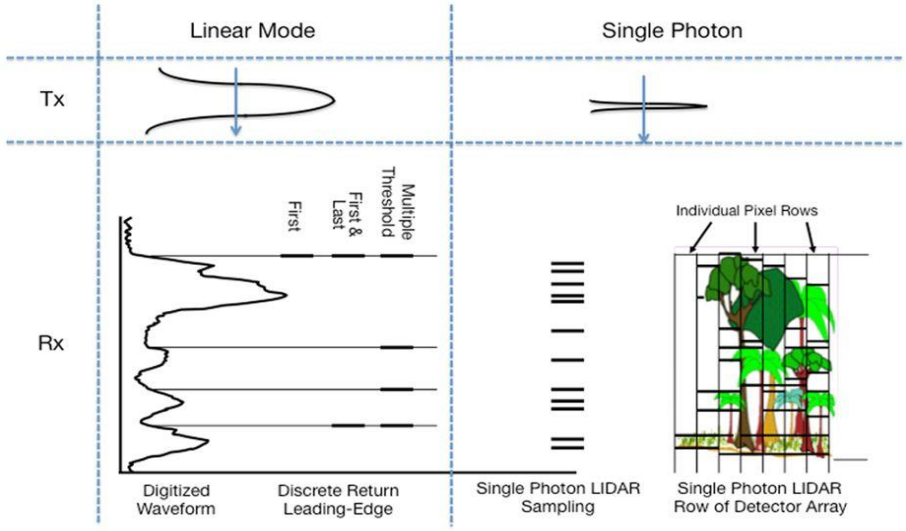


Figura 5: Diagrama esquemático que muestra la comparación entre los dos sistemas LiDAR. Tx es el pulso de láser transmitido y Rx es la energía devuelta. El pulso del láser SPL tiene un ancho de pulso más corto que otros sistemas. El detector consiste en una matriz de 10 x 10 que registra varios retornos por pulso. Fuente: Sirota et a

El resultado es que los sensores SPL pueden adquirir nubes de puntos de alta densidad de 12 a 30 puntos por metro cuadrado con muchos retornos por debajo de la vegetación. En este punto debemos comentar que la densidad de puntos al igual que en cualquier sistema LIDAR varía inversamente con la altura de vuelo. Si la altura de vuelo se duplica, la superficie cubierta se duplicará, pero la densidad de puntos se reducirá a la mitad. Un instrumento SPL volando a 200 nudos (370 km/h) a 4000 m sobre el nivel del suelo producirá una densidad de puntos de aproximadamente 20 puntos por metro cuadrado. (Sirota et al, n.d)

Este tipo de sensores en un primer momento se idearon para ser montados en satélites y operar desde el espacio. Concretamente el SPL100 el cual se utiliza en este trabajo fue el primer sensor LIDAR con tecnlogia SPL lanzado por *Leica Geosystems* al espacio. Esta tecnología fue desarrollada originalmente en colaboración con la NASA y en su primera misión se utilizó para medir desde el satélite ICESat-2, la elevación del hielo de la Tierra. (Gisresources, 2017)

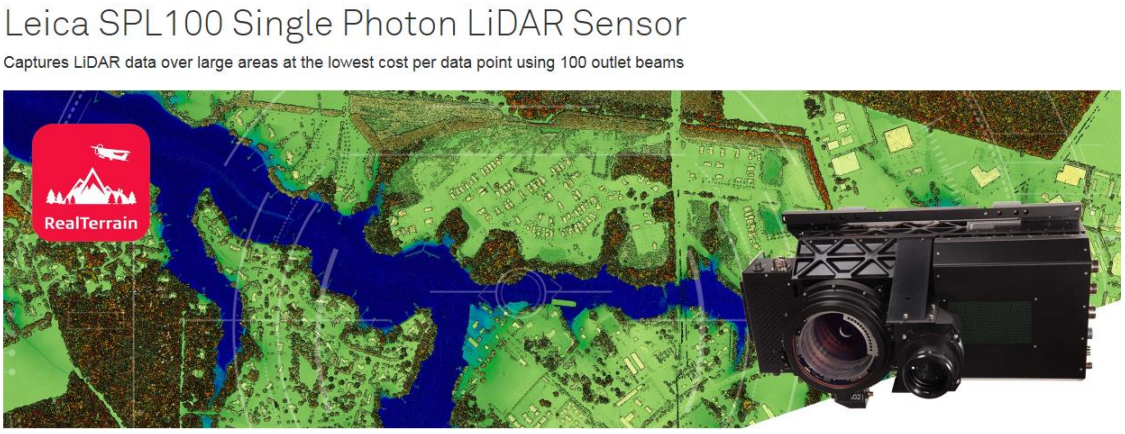


Figura 6: Sensor Leica SPL100. Fuente Leica Geosistems.

Los instrumentos SPL desarrollados hasta el momento operan en la longitud de onda verde (532 nm) en comparación con el infrarrojo cercano (1024 nm) utilizado por los instrumentos LiDAR convencionales (Harding, Dabney, & Valett, 2011). Esto se debe a que los detectores de un solo fotón en el infrarrojo cercano aún no están ampliamente disponibles. Mientras que la longitud de onda verde es beneficiosa para la batimetría y el mapeo de la capa de hielo, no es ideal para

estudios de vegetación. La longitud de onda de 532 nm es más sensible al ruido de fondo del sol y la reflexión de las hojas se reduce más en el visible que en el infrarrojo cercano. (Harding et al., 2011)

Debido a esto las nubes de puntos generadas con sensores SPL incluyen más retornos solares de fondo que otros instrumentos LiDAR cuando se vuelan en condiciones de luz diurna. Este hecho requiere un importante post-procesamiento de los datos y complica la recuperación de la estructura de la cubierta desde el SPL cuando los datos se adquieren en condiciones de luz solar intensa. (Degnan et al., 2007)

Como resultado, ha aumentado la preocupación por el alto nivel de ruido y la baja reflectancia en las longitudes de onda verdes, especialmente cuando la densidad de puntos es muy baja y las cubiertas son muy densas (Gwenzi & Lefsky, 2014). Por el contrario, las comparaciones de los datos de recuento de fotones a partir de las longitudes de onda visibles e infrarrojas han mostrado que la altura del dosel y otras mediciones de la estructura vertical eran similares en la mayoría de los casos, apoyando el uso de la longitud de onda verde SPL en la recuperación de la estructura del dosel si la densidad de muestreo es alta. (Gwenzi & Lefsky, 2014)

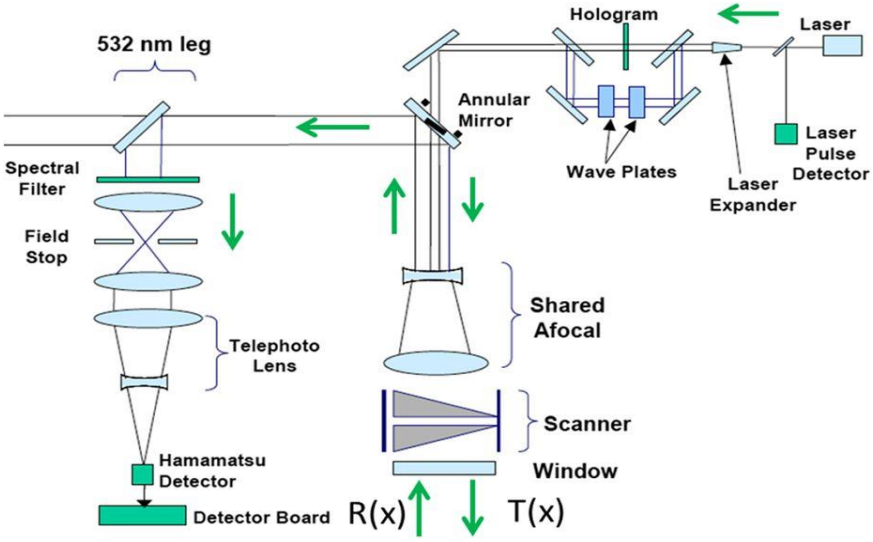


Figura 7: Diagrama esquemático de la cadena de componentes ópticos y del flujo de fotones en un sistema LIDAR SPL. Fuente: Swatantran et al., 2016

2.2 Métodos de clasificación de puntos LIDAR

Una vez presentada la tecnología LIDAR que se va a usar en este trabajo pasaremos a exponer los diferentes trabajos que se han realizado en clasificación LIDAR mediante minería de datos para posteriormente centrarnos únicamente en los que se han ocupado de la clasificación de líneas de alta tensión y otros cables.

2.2.1 Clasificación basada en minería de datos

En esta primera revisión bibliográfica sobre clasificación LIDAR se expondrán trabajos de clasificación de datos LiDAR mediante minería de datos. Aunque son muchos los trabajos realizados de clasificacion en datos LIDAR con mineria de datos en esta primera parte hemos seleccionado algunos que consideramos relevantes para entender este trabajo. Niemeyer et al (Niemeyer, Rottensteiner, & Soergel, 2013) proponen la clasificación de datos LIDAR mediante Random Forest. Este algoritmo ha demostrado que realiza una clasificación correcta en datos

LIDAR. Otros autores también proponen el uso de este algoritmo para la clasificación de datos LIDAR donde se debe realizar la clasificación de múltiples clases (Chehata, Guo, & Mallet, 2009).

Otro algoritmo que generalmente es usado en la clasificación de datos LIDAR mediante minería de datos es SVM (*Support Vector Machine*). Li et al (Li, Liu, Walker, Hayward, & Zhang, 2010) proponen un método de clasificación basado en este algoritmo para la clasificación de datos LIDAR en zonas urbanas. Comentar que en este estudio los autores emplearon además un algoritmo de balanceo para eliminar valores atípicos de los datos LIDAR (Zhang, Lin, & Ning, 2013). Esto se deberá tener en cuenta también en nuestro trabajo de clasificación de líneas de alta tensión. Estos autores utilizaron tanto variables geométricas, radiométricas, topológicas y las correspondientes a las características de los retornos. Con estas variables hicieron el uso de máquinas de soporte vectorial (SVM). De acuerdo a las conclusiones obtenidas por los autores anteriores sugieren que SVM realiza una clasificación de datos LIDAR con una precisión global de clasificación mayor que 92,34 % y que esta precisión aumenta con el aumento de la densidad de puntos. Estos datos serán tenidos en cuenta para la realización del presente trabajo.

En otro trabajo consultado los autores(Collin, Archambault, & Long, 2011), proponen el empleo de técnicas de aprendizaje automatizado en la clasificación de ecosistemas marinos utilizando diferentes algoritmos de clasificación. Estos autores realizan la clasificación utilizando *Naive Bayes*, arboles de decisión, C4.5, *Random Forest*, SVM y CN2, y compararon los resultados para la clasificación de ocho especies capturadas con LIDAR batimétrico. En este trabajo tras comparar los diferentes algoritmos llegan a la conclusión que el algoritmo que más destaco en la clasificación fue el algoritmo implementado en la clasificación a partir de *Random Forest* (Collin et al., 2011). En este trabajo se realizará un proceso parecido de comparación de diferentes algoritmos para la detección de líneas de alta tensión.

Por último en la clasificación de datos LIDAR en general se hablara en este apartado del trabajo realizado por (Garcia-Gutierreza, Concalves-Seco, & Riquelme-Santos, 2009) de la Universidad de Sevilla. Estos autores proponen una clasificación mediante el empleo de algoritmos como *Naive Bayes* y árboles de decisión C4.5, obteniendo en su clasificación mejor beneficio en la precisión con el uso del algoritmo de árbol de decisión C4.5 (Garcia-Gutierreza et al., 2009).

2.2.1. Detección y clasificación de cables a partir de datos LIDAR

Una vez que hemos expuesto algunos trabajos realizados con algoritmos de minería de datos para la clasificación de datos LIDAR con las que se va a realizar el trabajo nos centraremos ahora en conocer los trabajos realizados en la detección de cables a partir de datos LIDAR. En los trabajos consultados que se han realizado hasta ahora generalmente el problema de la identificación y clasificación de las líneas eléctricas se ha realizado a partir de imágenes aéreas de forma manual o de forma semiautomática (Sohn, Jwa, & Kim, 2012)(Yang, Wei, Li, & Li, 2012).

En esta clasificación se suele diferenciar los trabajos realizados en zonas abiertas donde las líneas eléctricas son visibles en imágenes aéreas donde generalmente la clasificación es fácil. Sin embargo como hemos podido consultar en la bibliografía, en las áreas forestales, esta tarea resulta más complicada (Zhu & Hyyppä, 2014).

Desde la llegada de los sensores LiDAR se ha intentado realizar este trabajo a partir de datos LIDAR de una forma más automática y además más económica. En la bibliografía consultada se han encontrado numerosos trabajos que intentan clasificar líneas de alta tensión a partir de datos LIDAR. Sin embargo, el procesamiento y clasificación de estos datos para la extracción de las líneas de alta tensión y cables sigue desarrollándose de forma manual (Kim & Sohn, 2011).

De acuerdo a las trabajos consultados desde hace años ya se realiza la clasificación de cables a partir de datos LIDAR, pero este proceso se realiza en gran medida de forma manual (Vosselman

& Maas, 2010). Por lo tanto, podemos decir que se necesitan métodos que realicen este trabajo automáticamente. Según algunos autores gracias al formato de los datos LIDAR los objetos de diámetro pequeño presentan retornos y es posible discriminar cables gracias a ello (Melzer & Briese, 2004). Esto ha sido posible gracias a los avances en los sensores LIDAR, y más concretamente en el aumento de la densidad de puntos por m2. Según hemos podido leer actualmente se ha llegado a una densidad de aproximadamente 60 puntos por m2 lo que hace que la clasificación de líneas de alta tensión sea realizable de forma más precisa. (Zhu & Hyyppä, 2014)

Aunque la mayoría de trabajo consultados detectan cables en datos con alta densidad de puntos por m2, también hemos encontrado artículos donde se detectan cables con menos densidad de puntos, (Clode & Rottensteiner, 2005) detectaron árboles y líneas eléctricas a partir de menos de un punto por m2 de nube de puntos en Sydney. Estos autores lo realizaron utilizando solo las diferencias entre el primer y el último retorno. Podemos ver los resultados en la figura 8.

Figura 8: Vegetación (verde) y cables (negro) clasificados mediante diferencias entre primer y último retorno. Fuente Clode et al.

En trabajos anteriores también (Axelsson, 1999) se utilizaron datos LiDAR con una densidad de 8 puntos/m2 para la detección de líneas eléctricas. Este autor presentó un método basado en la búsqueda de estructuras 2D paralelas y lineales mediante la transformada de *Hough*. (cita trabajo) En la tabla 4 presente al final de la revisión bibliográfica se observarán los resultados de todos los trabajos consultados. Otros autores también utilizaron este método para realizar la clasificación de datos LIDAR (Melzer & Briese, 2004) con una posterior transformación a 3D mediante métodos de ajuste de curvas.

En trabajos posteriores donde la densidad de puntos aumento hasta unos 10 puntos/m2 se (McLaughlin, 2006) utilizaron datos LiDAR con una distancia media entre puntos de 1,2 m-2,4 m para la detección y clasificación de líneas eléctricas. Este autor utilizo un método de clasificación supervisado con el que consiguió unos resultados del 72% de líneas eléctricas clasificadas correctamente.

Otros autores (Wang et al., 2017) utilizaron la técnica de detección líneas eléctricas con una imagen de nivel de gris 2D, la intensidad del retorno del láser y una transformación mejorada de Hough. (Zhu & Hyyppä, 2014)

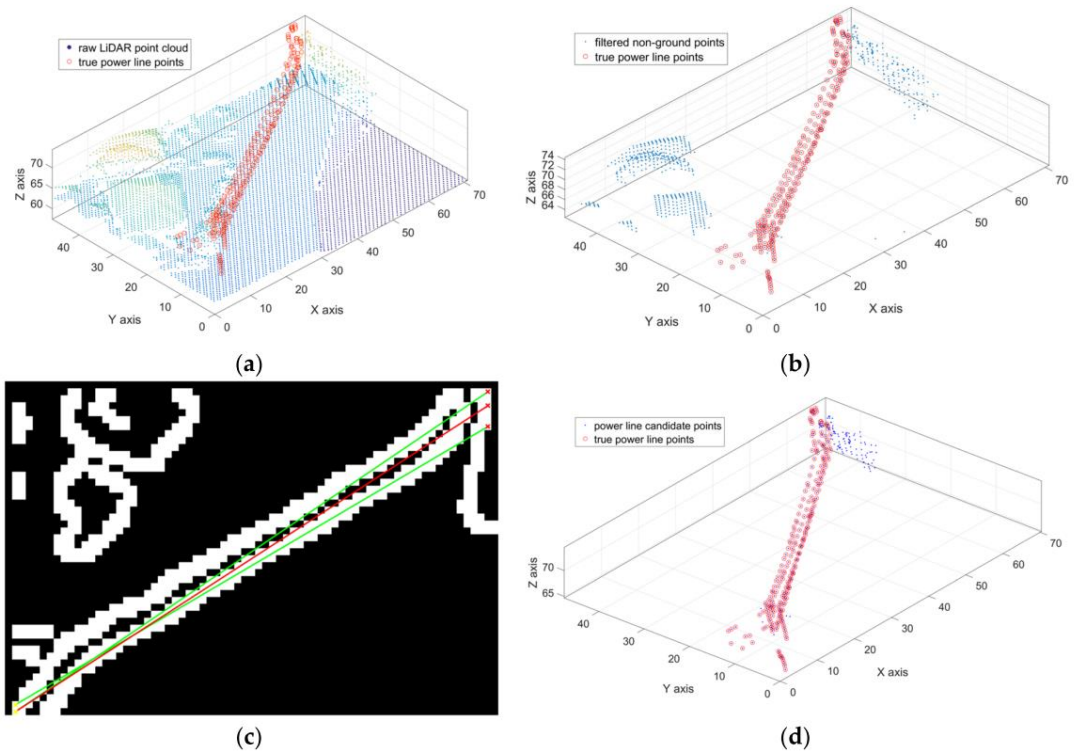


Figura 9: Visualización del proceso de filtrado de candidatos para líneas eléctricas. (a) Nube de puntos LiDAR en bruto; c) la dirección del corredor 2D de la línea eléctrica detectada en el plano XOY. Los puntos verdes son extraídos por la transformación de Hough y la línea roja es la dirección del cable de la línea de energía construida por el algoritmo RANSAC; d) los puntos candidatos a la línea de potencia filtrados por la dirección del pasillo de la línea de potencia: los los puntos rojos son los verdaderos puntos de línea eléctrica, los azules son los resultados de filtrado del candidato a línea eléctrica. Fuente: Wang et al., 2017

En el artículo consultado de Jwa et al (Y Jwa et al., 2009) observamos un método basado en VPLD (*Voxel PowerLIne detector*) para la reconstrucción automática de la línea eléctrica utilizando una densidad de 5 puntos/m2 de datos LiDAR. En otros muchos trabajos consultados se (Kim & Sohn, 2011) utilizo tambien el método de *random sample consensus* (RANSAC) para determinar qué puntos pertenecen a una línea junto con la mínima distancia y el análisis de componentes principales en la extracción de características y *Random forest* como algoritmo de clasificación con una densidad de 30 puntos por m2.

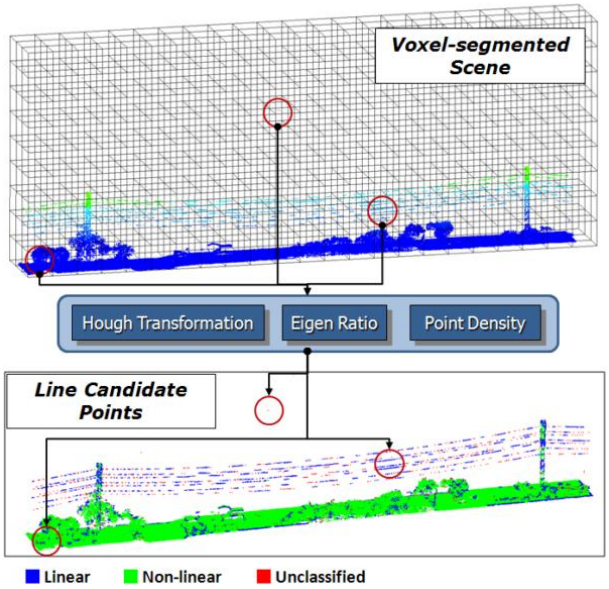


Figura 10: Ilustración de una clasificación basada en vóxeles 3D de los puntos candidatos para líneas de alta tensión. Fuente: Y Jwa et al, 2009

Los trabajos anteriores que hemos visto se pueden incluir en dos grandes grupos, uno en los que los métodos tratan de buscar elementos lineales (RANSAC y 2D *Hough transformation*) y otros que tratan de clasificar los cables mediante algoritmos de clasificación supervisada. En cuanto a los métodos de detección basados la búsqueda de elementos lineales hemos observado que requieren de un costo computacional alto, especialmente para un conjunto de datos de gran tamaño (Zhu & Hyyppä, 2014). Este alto coste computacional se debe a que estos deben ir punto por punto para determinar si pertenecen o no a una línea. Algunos investigadores han propuesto la realización de estos métodos por cuadrantes (Yoonseok Jwa & Sohn, 2012).

En cuanto a los métodos de clasificación supervisados todos los autores consultados están de acuerdo en que se requiere un gran conjunto de datos de entrenamiento para lograr los resultados deseados. Además, el muestreo desequilibrado dará lugar a una mayor tasa de errores de clasificación (Zhu & Hyyppä, 2014).

Además de estas técnicas comentadas hasta ahora debemos hablar también de los estudios que utilizan el escáner láser móvil (MLS). Esta técnica ha sido utilizada por (Zhu & Hyyppa, 2014) para la cartografía de líneas eléctricas la cual ha resultado muy relevante. La idoneidad de la extracción de objetos a partir de sensores LiDAR aerotransportados y sensores móviles terrestres se comenta también en el citado estudio. Con lo que respecta a la extracción de líneas de alta tensión, el sistema terrestre es adecuado para la cartografía de áreas pequeñas, pero que requieren un alto nivel de detalle. Utilizar esta técnica para el mapeo de áreas grandes presenta un alto coste computacional debido a la alta densidad de puntos de MLS. Sin embargo, con sensores aerotransportados no sólo se puede cubrir una gran área, sino que también es adecuada para terrenos ondulados (por ejemplo, colinas o montañas) que son difíciles de alcanzar utilizando vehículos terrestres (Zhu & Hyyppä, 2014).

En la tabla 4 podemos ver un resumen con los estudios mencionados en la revisión bibliográfica y el porcentaje de fiabilidad obtenida, así como la técnica de clasificación utilizada y algunas observaciones al proceso.

Cita	**Técnica de clasificación**	**Entorno**	**Fiabilidad obtenida**	**Observaciones**
Zhu e Hyyppa	Análisis estadístico basado en criterios de métricas como la altura, la densidad y el procesamiento basado en imágenes 2D que considera las propiedades geométricas	Forestal	93.26%	
Clode y Rottensteiner	Técnica basada en diferencias entre primer y último retorno	Forestal	64-72%	Obtuvieron mejores resultados en detectar vegetación que en líneas eléctricas.
Cheng et al	Sistema jerárquico basado en vóxeles y un método de filtrado ascendente	Urbano	93%	Alta densidad de puntos.
Sohn et al	Sistema de clasificación supervisada para extraer puntos de candidatos lineales y luego convertirlos en segmentos de línea mediante (*RANSAC*)	Urbano y Forestal	91.30%	
Melzer y Briese	Transformación Hough para detectar la potencia segmentada en 2D	Forestal	n.d	La reconstrucción de las líneas eléctricas se llevó a cabo utilizando el muestreo aleatorio
Kim y Sohn	*RANSAC* y análisis de los componentes principales en la extracción de características y *Random forest* como técnica de clasificación.	Urbano	91%	Los métodos no eran adecuados en escenas urbanas complejas en las que existían muchos postes eléctricos pequeños.
Axelsson	Búsqueda de estructuras 2D paralelas y lineales basadas en el método de transformación de Hough.	Forestal	n.d	
McLaughlin	Método de clasificación supervisada.	Forestal	72%	
Wang et a	Imagen de nivel de gris 2D, la intensidad del retorno del láser y una transformación mejorada de *Hough*.	Urbano	98%	
Y Jwa et al	Método basado en VPLD (*Voxel PowerLIne detector)*	Urbano y Forestal	92%	

Tabla 4: Tabla resumen de las trabajos consultados sobre detección de cables. Fuente: Elaboración propia.

2.3 Introducción a la minería de datos

La **minería de datos** o **KDD** (*Knowledge Discovery in Databases*) es una tecnología que permite a partir de un proceso de análisis descubrir conocimiento dentro de grandes volúmenes de datos (Sanz-Delgado, 2018). El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior (Perdomo, 2007).

Actualmente el término minería de datos es una palabra de moda, y es frecuente verla mal utilizada para referirse a cualquier forma de procesamiento de la información o *big data* y también se ha generalizado a cualquier tipo de sistema de apoyo informático de decisión, incluyendo la inteligencia artificial, aprendizaje automático y la inteligencia empresarial (Perdomo, 2007). La tarea de minería de datos real es el análisis automático o semi-automático de grandes cantidades de datos para extraer patrones interesantes hasta ahora desconocidos (Perdomo, 2007).

Podemos decir por tanto que la minería de datos debe ser considerada como un área multidisciplinaria que relaciona procedimientos, métodos, modelos y técnicas provenientes de la estadística, reconocimiento de patrones, del aprendizaje de máquinas, reconocimiento de patrones y las bases de datos (Perdomo, 2007). Debido a estas características son muchos los ámbitos de aplicación de la minería de datos. Actualmente estos métodos se aplican tanto en problemas de medicina como finanzas, comercio, seguridad, geografía y cualquier otro espacio científico del que se disponga de gran cantidad de datos de los que extraer patrones (Sanz-Delgado, 2018).

Lo que tienen en común todas estas disciplinas en el proceso de extracción de conocimiento al igual que este trabajo de investigación es que se trata de procesos automáticos compuestos por los mismos pasos. Siempre debemos poseer el conjunto de datos de entrenamiento que permite extraer el conocimiento, el modelo que se implementa a través de un programa y la validación del mismo. Por lo tanto, en todos los procesos de extracción de conocimiento a partir de los datos se dan los siguientes procesos (Figura 11) (Sanz-Delgado, 2018).

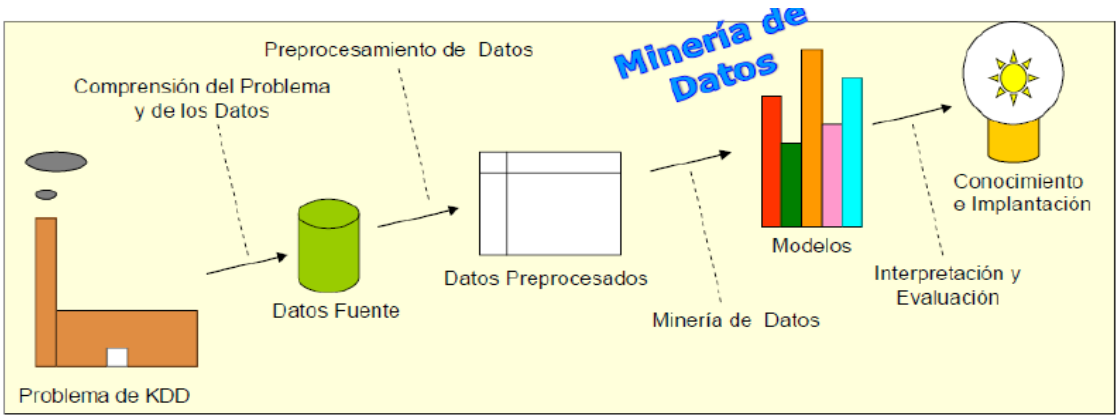


Figura 11: Esquema de trabajo en minería de datos. Fuente: Sanz-Delgado

En cuanto a los procesos de extracción de conocimiento se puede decir que en la fase de preparación de los datos suele ser necesario la realización de la limpieza, reducción y transformación de los datos. Generalmente para encontrar el modelo definitivo se realizan muchos ensayos hasta que se encuentra el que mejores resultados de extracción de conocimiento presenta. El último paso de la minería de datos y el que nos indica la calidad del modelo es la aplicación a otros datos y evaluación de la predicción realizada. (Sanz-Delgado, 2018)

Otra parte importante a tener en cuenta en las fases de la minería de datos es comentar que generalmente en la preparación de los datos es donde más tiempo y esfuerzo se ha de dedicar (Figura 12) ya que de una buena selección y preparación de los datos dependerá un resultado satisfactorio de los modelos de minería de datos obtenidos (Sanz-Delgado, 2018).

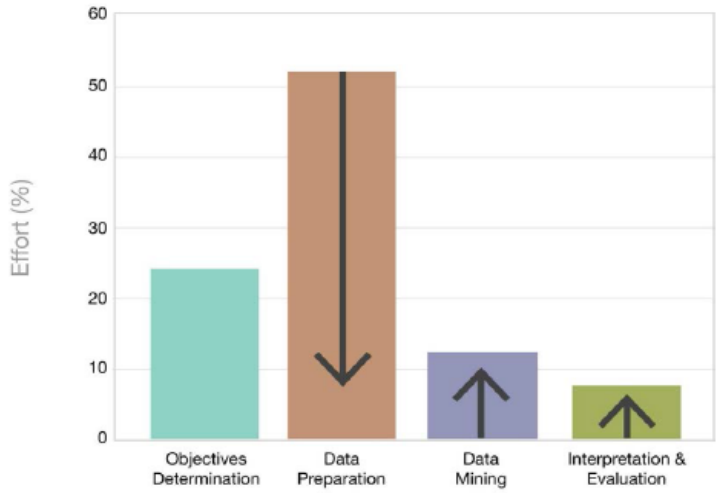


Figura 12: Tiempo y esfuerzo en cada fase del proceso de minería de datos. Fuente: Sanz-Delgado

Para la realización del modelo se utilizan algoritmos, éstos se clasifican en supervisados o predictivos y no supervisados o descriptivos, aunque últimamente se están considerando también los semi-supervisados. Dentro de los métodos supervisados o predictivos se encuentran aquellos que usan un conjunto de datos a modo de entrenamiento y dentro de los no supervisados a los basados en el ajuste del modelo a las observaciones disponibles.

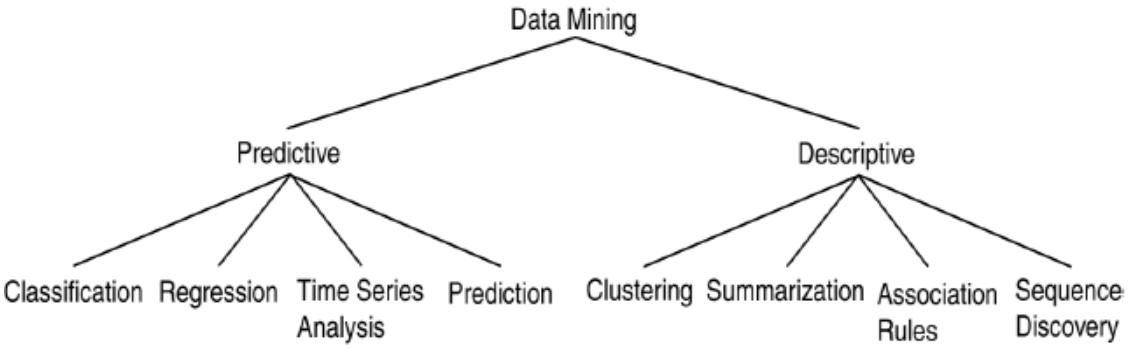


Figura 13: Tipos de algoritmos más utilizados en minería de datos. Fuente: Sanz-Delgado

En los modelos no supervisados o descriptivos no se conoce el valor de la variable a predecir, mientras que en los modelos supervisados se conoce el valor de la variable a predecir mediante el uso de datos de entrenamiento. En el caso de los métodos semi-supervisados se usan tanto datos de entrenamiento etiquetados como no etiquetados. (Witten & Frank, 2005)

En este trabajo se han utilizado técnicas de clasificación supervisadas y no supervisadas para evaluar la calidad de ambos tipos de modelos. Los diferentes modelos que se usan en este trabajo y que se explicaran brevemente en esta parte del trabajo son los siguientes:

1. Arboles de decisión
    a. C4.5
    b. CART
2. Métodos basados en Multiclasificadores (*Ensembles*)
    a. Random Forest
    b. Bagging
    c. AdaBoost
3. Algoritmos del tipo vecino más cercano (*Nearest Neighbors*)
    a. KNN

2.3.1 Árboles de decisión.

Un árbol de decisión es un clasificador que en función de un conjunto de atributos permite determinar a qué clase pertenece el caso objeto de estudio, representa un conjunto de decisiones organizadas en una estructura jerárquica (Sanz-Delgado, 2018). Los árboles de decisión proveen de una herramienta de clasificación muy potente. Su uso en el manejo de datos la hace ganar en popularidad dadas las posibilidades que brinda y la facilidad con que son comprendidos sus resultados (Perdomo, 2007).

El objetivo de un árbol de decisión es crear un modelo que prediga los valores de una variable mediante la entrada de otras variables distintas a través de una base de datos. A partir de esta base de datos se buscan patrones para decidir a qué clase pertenecerán los nuevos datos a los que se enfrentara el árbol de decisión, para ello se dividen las áreas de entrenamiento en un conjunto de entrenamiento y un conjunto de test con el que se prueba la calidad del modelo (Martínez Blanco, 2016).

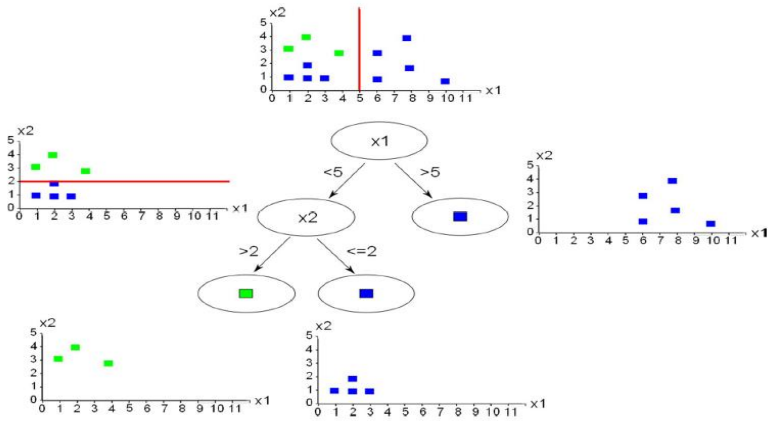


Figura 14: Esquema trabajo de un árbol de decisión. Fuente: Sanz-Delgado

La estructura de un árbol de decisión está constituida por nodos, que pueden ser hojas donde se toma la decisión, unidos entre sí por ramas que se etiquetan con los valores de los atributos, intentando buscar a través de las distintas clases los valores del atributo con el que se etiqueta al nodo padre. Estos nodos se pueden unir con otros nodos o con las hojas a través de las ramas, estableciendo las hojas los distintos valores del atributo a predecir, motivo por el que las hojas aparecen al final del árbol (Sanz-Delgado, 2018). Podemos ver un ejemplo de un árbol de

decisión en la figura 14. Generalmente los árboles de decisión sirven para clasificar los atributos en clases o lo que es lo mismo valores discretos. Si los valores son continuos entonces se deben usar los árboles de regresión. (Witten & Frank, 2005)

En cuanto a la forma de construir los arboles de decisión existen dos formas, de inducción: de abajo a arriba y de arriba abajo, siendo estos últimos lo más utilizados por la literatura dando lugar a lo que se denomina arboles con la técnica de "divide y vencerás". En este último tipo de árboles el atributo más importante se ubica en la parte superior del árbol y en función de los valores de los atributos iterativamente se van realizando las distintas particiones del conjunto de entrenamiento según una regla de división que mide lo adecuado que resulta una variable para constituir un árbol (Martínez Blanco, 2016).

En este trabajo se utilizarán distintos métodos de selección de atributos en cada nodo: por un lado, se utiliza la ganancia de información, basada en la entropía, que da lugar al árbol de decisión C4.5 y por otro lado el índice GINI que da lugar a CART. Ambos realizan un proceso de poda para prevenir el sobre-aprendizaje.(Garcia-Gutierreza et al., 2009).

En ambos tipos de árboles la fase de clasificación o entrenamiento se detiene cuando se cumple alguno de los siguientes criterios (Sanz-Delgado, 2018):

- Todos los ejemplos del entrenamiento pendientes pertenecen a la misma clase.
- El número de ejemplos de un nodo es menor que un umbral dado.
- El número de ejemplos que se derivan a una rama es menor que un umbral dado.

La poda de los árboles de decisión pretende evitar que el árbol de decisión se sobre-ajuste al conjunto de entrenamiento, de manera que se cortan los árboles sobre-ajustados dejándolos en otros más pequeños tras quitarles las ramas que no contribuyen a generalizar la precisión (Alexander, Tansey, Kaduk, Holland, & Tate, 2011)(Martínez Blanco, 2016).

2.3.2 Métodos basados en Multi-clasificadores (*Ensemble*)

Los métodos basados en multi-clasificadores o *ensemble* son un tipo de modelo de decisión que se basan en construir un modelo de aprendizaje mediante un conjunto de clasificadores que mejore el resultado de cada clasificador por separado (Scikit-learn, 2017).

Este tipo de modelos o algoritmos al igual que la mayoría necesita un conjunto de datos de entrenamiento que contienen los datos etiquetados pero además necesita un algoritmo de inducción o algoritmo base que a partir del conjunto de entrenamiento genera una clasificación que representa la relación generalizada entre los atributos de entrada y la variable a predecir, el generador de diversidad para crear las distintas clasificaciones y el que combina para juntar las distintas clasificaciones (Rokach & Maimon, 2008).

En general, se trata de algoritmos que van buscando reducir el sesgo y la varianza en el aprendizaje supervisado. Estos métodos se clasifican en dos grandes grupos Métodos de promedio (*averaging methods*) y Métodos de impulso (*boosting methods*) (Martínez Blanco, 2016). En este trabajo se utilizará un tipo de modelo de cada grupo y además un método que combina ambos grupos de algoritmos. Estos modelos serán explicados a continuación.

2.3.2.1 Bagging

El primer algoritmo de multi-clasificación que vamos a utilizar en el trabajo pertenece a los métodos de promedio o independientes: En la forma de clasificar de estos algoritmos se construyen distintos modelos independientes y como resultado final se elige el de mayor voto. En este trabajo el modelo que vamos a utilizar de estos métodos es el denominado *bagging* (***b**oostrap **agg**regat**ing***).

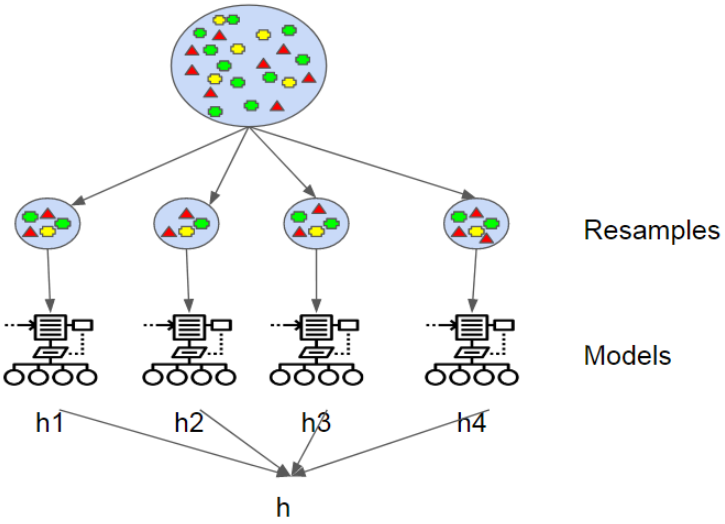


Figura 15: esquema de trabajo de clasificador Bagging. Fuente: hackernoon.com

El ***bagging*** constituye un método basado en multi-clasificadores que construye un clasificador utilizando nuevos subconjuntos de entrenamiento todos ellos del mismo tamaño y menor que el del conjunto original ofreciendo como solución una predicción con una varianza menor que la obtenida por cada uno de los subconjuntos, evitando el sobre-ajuste (Breiman, 1996).

2.3.2.2 Adaboost

El segundo algoritmo basado en métodos de multi-clasificadores utilizado en este trabajo va a ser un algoritmo del tipo métodos de impulso o dependientes llamado ***AdaBoost***. Este método se puede aplicar a los clasificadores del tipo árboles de decisiones, para mejorar la fiabilidad, el rendimiento y la resistencia al sobre entrenamiento. En este algoritmo el árbol de decisión único es sustituido por un grupo de árboles. Cada árbol de decisión se denomina clasificador débil, cada clasificador débil es entrenado de manera iterativa para mejorar sobre el anterior (Marsh, 2016).

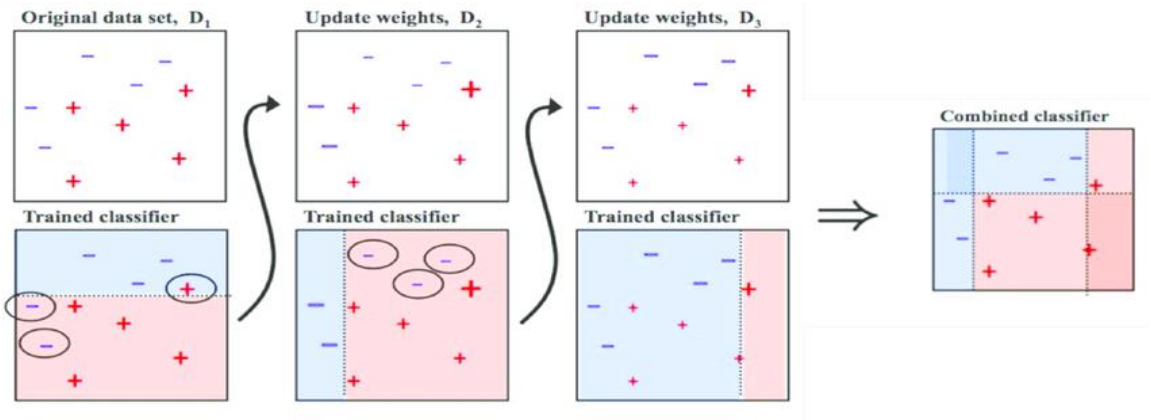


Figura 16: Esquema de trabajo de clasificador Adaboost. Fuente:(Marsh, 2016)

El primer clasificador es entrenado como un árbol de decisión único a partir de los datos del anterior ponderando la importancia de los datos de la formación para el siguiente (Figura 16). Los puntos que se clasificaron correctamente reciben pesos pequeños, mientras que los puntos clasificados incorrectamente reciben pesos grandes. De esta manera, cada árbol del multi-clasificador se centrara en los puntos que no han sido bien clasificados por el anterior clasificador (Marsh, 2016).

Indistintamente de que se trate de métodos dependientes o independientes, por lo general al combinar varios modelos se consiguen mejores resultados, reduciendo la varianza y evitando el sobre-ajuste (Martínez Blanco, 2016). Esto se va a conseguir con el tercer algoritmo utilizado en el trabajo que se conoce como *Random Forest* y se explicara en el siguiente punto.

2.3.2.3 Random forest

El método ***Random Forest*** constituye un algoritmo de aprendizaje supervisado que se encuentra dentro de los métodos de multi-clasificadores. Este método combina la idea de *Bagging* propuesta por Breiman 1996 y la de *Random Subspace* (RS) de Ho 1998 (Breiman, 1996) (Ho, 1998). En este método se utiliza una técnica en la que para crear los distintos modelos se introducen alteraciones al azar en el método de aprendizaje (Sanz-Delgado, 2018). Estas alteraciones aleatorias son una combinación de pronósticos de árboles de decisión de tal manera que cada árbol depende de los valores de una variable tomados aleatoriamente. El error de generalización de este método depende de la cantidad de votos de los árboles de decisión individuales y la correlación entre ellos (Breiman, 2001).

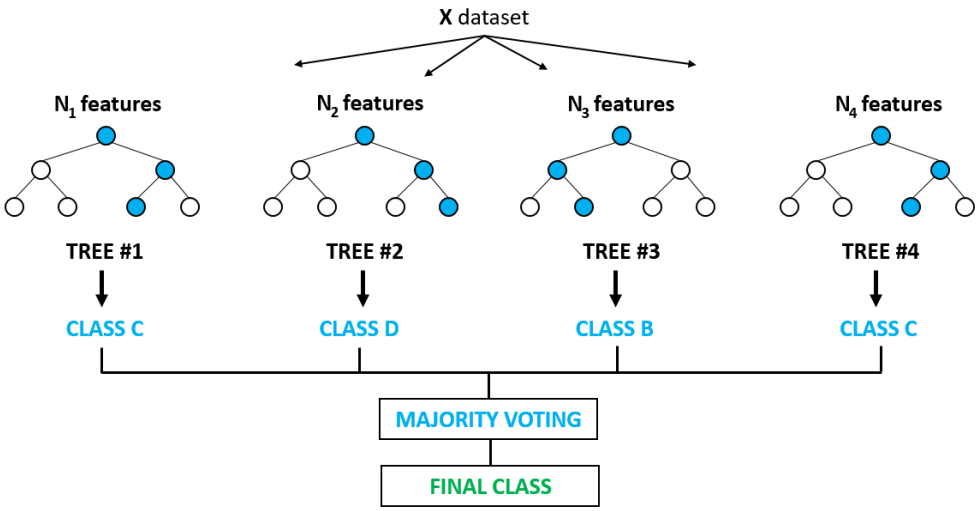


Figura 17: Esquema de trabajo de Random Forest. Fuente: http://www.globalsoftwaresupport.com/random-forest-classifier-bagging-machine-learning/

Entre las ventajas que se le atribuyen al método *Random Forest* destacan:

- Resulta eficiente con grandes bases de datos.
- Puede tratar cientos de variables sin excluir ninguna.
- Ofrece una estimación de las variables más importantes.
- Permite crear múltiples árboles de decisión de manera paralela.

Como desventajas se suelen indicar principalmente que si en los datos existe ruido el algoritmo se sobre-ajusta y que las clasificaciones realizadas por *Random Forest* resultan difíciles de interpretar (Martínez Blanco, 2016). Además, entre los inconvenientes encontramos que es una técnica sensible al desequilibrio de la muestra de entrenamiento. Es decir, en caso de problemas no equilibrados (clases con número de muestras de training muy distintas) tiende a favorecer la asignación a las clases más frecuentes (Jes, 2016).

2.3.3. Vecino más cercano

El método de **vecino más cercano** es un método de clasificación no supervisado basado en los puntos vecinos. Este método es la base de muchos otros métodos de aprendizaje, en particular el aprendizaje múltiple y espectral (Scikit-learning, 2017). El principio detrás del método del vecino más cercano es encontrar un número predefinido de muestras de entrenamiento más cercanas al nuevo punto y predecir la etiqueta a partir de ellas.

El número de muestras puede ser una constante definida por el usuario (*k-nearest neighbor learning*), o puede variar en función de la densidad de puntos (*radius-based neighbor learning*). La distancia puede ser cualquier medida métrica: la distancia estándar euclidiana es la opción más común. Los métodos basados en vecinos son conocidos como métodos no generalizadores de aprendizaje automático, ya que simplemente "recuerdan" todos sus datos de entrenamiento y asignan el nuevo dato de entrada al que más se asemeje de su conjunto de entrenamiento (Scikit-learning, 2017).

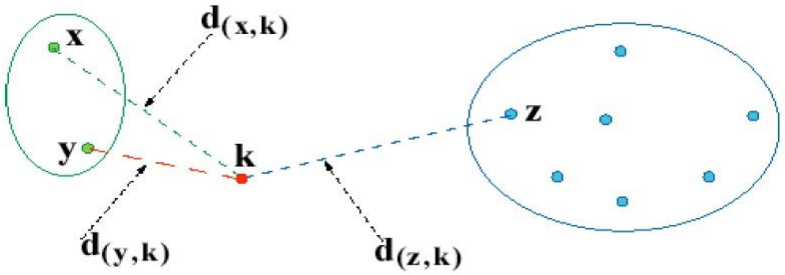


Figura 18: Esquema de trabajo de los algoritmos basados en vecino más cercano. Fuente:

A pesar de su simplicidad estos métodos han tenido éxito en un gran número de problemas de clasificación incluyendo dígitos manuscritos o escenas de imágenes satelitales. Al ser un método no paramétrico, a menudo tiene éxito en situaciones de clasificación en las que el límite de decisión es muy irregular (Jes, 2016). Por estas razones se ha decidido incluir uno de estos métodos en este trabajo de investigación.

3- MATERIAL Y MÉTODOS

3.1 Área de estudio

Los datos empleados en este trabajo tanto para la creación de las áreas de entrenamiento y test como los datos a clasificar posteriormente son los datos obtenidos en el vuelo LiDAR realizado en Navarra en 2017.

En esta parte del trabajo se expondrá brevemente el proyecto de Vuelo LiDAR, así como las características y los datos resultantes de este vuelo concreto. Como hemos podido leer en la revisión bibliográfica las características de los datos LiDAR como la densidad de puntos y el tipo de tecnología utilizada en su captura puede tener una gran influencia en la calidad de los resultados obtenidos en la clasificación y en el tipo de algoritmo a utilizar para mejorar esta clasificación.

3.1.1 Vuelo LIDAR

El vuelo LiDAR se realizó en Navarra en el verano de 2017, las características de este vuelo fueron peculiares por el tipo de orografía que presenta el territorio. En un primer momento se dividió el ámbito del proyecto en una serie de pasadas, pero posteriormente hubo que realizar más pasadas complementarias para cubrir todo el terreno del proyecto. Esto fue debido a que los sensores LIDAR tienen un rango de altitud en el que los puntos son devueltos al sensor, por encima y debajo de ese rango se producen zonas sin datos. Debido a esto tras la realización del primer vuelo LIDAR en todo el territorio se observaron muchas zonas que presentaban vacíos de datos.

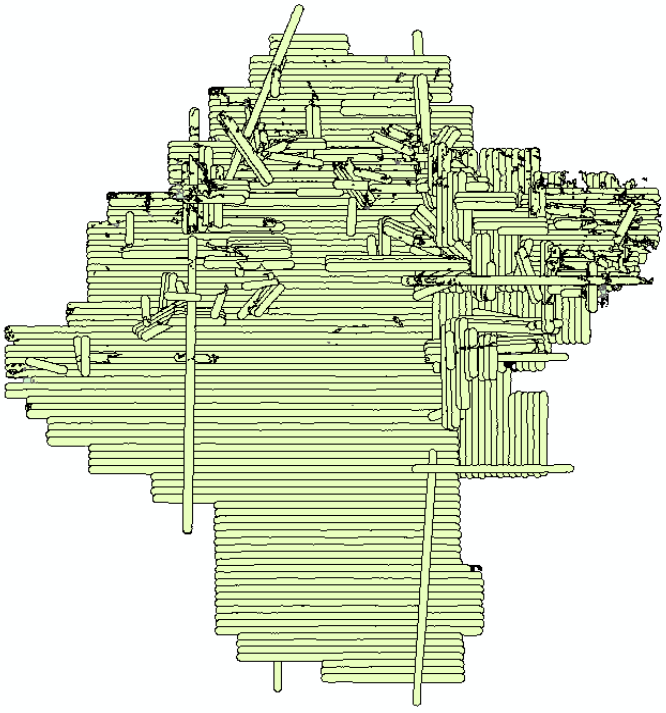


Figura 19: Pasadas realizadas en el vuelo LIDAR del proyecto. Fuente: Elaboración propia.

Para solucionarlo se debieron realizar multitud de pasadas en las zonas con una orografía más escarpada donde quedaban zonas sin datos debido a las diferencias de altitud entre los puntos más altos de la huella y los más bajos. Esto se observa en la diferencia del número de pasadas (figura 19) que se realizaron en unas zonas y en otras para alcanzar los objetivos de densidad de puntos contenida en el pliego de condiciones técnicas del proyecto. Como se puede observar en la imagen de las pasadas del proyecto de vuelo existen zonas con un gran número de pasadas.

La diferencia de pasadas entre unas zonas y otras hace que la densidad de puntos por metro cuadrado sea muy diferente según las pasadas realizadas en la zona (Figura 20). Como se puede observar en ambas imágenes en las zonas montañosas del territorio fueron necesarias un gran número de pasadas para conseguir un recubrimiento total del territorio mientras que en la zona sur se cumplió con el recubrimiento total realizando solo las pasadas previstas en el inicio del proyecto.

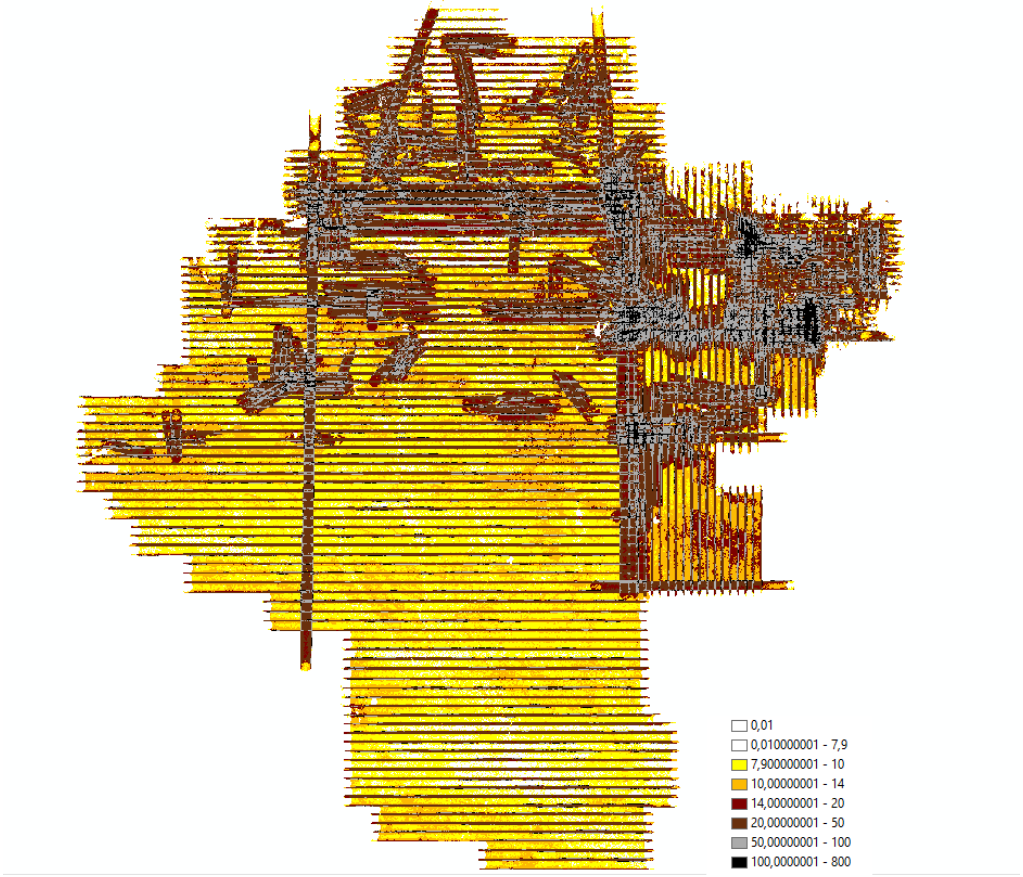


Figura 20: Densidad de puntos LiDAR por metro cuadrado en el ámbito del proyecto. Fuente: Elaboración propia.

Gracias a estas complicaciones y a la gran cantidad de pasadas solapadas en la zona noreste de Navarra debido a las diferencias altitudinales que comentábamos podemos obtener zonas con una densidad de puntos cercana a 60 puntos/m2 y otras zonas con una densidad media de 10 puntos por metro cuadrado. Estas diferencias de densidad entre zonas pueden ser beneficiosas a la hora de realizar las áreas de aprendizaje para entrenar al algoritmo ya que podremos obtener zonas de aprendizaje que presenten una gran cantidad de puntos pertenecientes a cables para un posterior mejor entrenamiento del algoritmo. Además, en la fase de clasificación podremos observar cómo se comporta el algoritmo en zonas con diferente densidad de puntos e intentar adaptarlo para que sea capaz de detectar cables con diferentes densidades de puntos/m2.

La forma en la que se disponen los datos para la realización del presente trabajo son bloques cuadrados en formato LAS con un tamaño de un kilómetro de lado. Debido a lo que comentábamos en el punto anterior el número de puntos de cada bloque LAS varía entre aproximadamente 4.000.000 y 60.000.000 de puntos dependiendo de la zona del proyecto seleccionada. Igualmente, el tamaño de los ficheros también varía entre aproximadamente 400MB los más pequeños y hasta archivos de 5GB los que presentan una densidad de puntos mayor.

3.1.2 Características de los datos LiDAR de este proyecto

Antes de seguir explicando la realización del trabajo debemos comentar unas características de los datos utilizados en este trabajo que complican aún más la correcta detección de líneas de alta tensión en estos datos LIDAR. Como se ha comentado en la revisión bibliográfica una de las características de la tecnología SPL LiDAR es la gran cantidad de ruido que presentan. La empresa que se ocupó de la realización del vuelo realizo una primera clasificación a gran escala de los datos y los entregó en una primera versión que no contenía los puntos correspondientes al ruido.

Tras observar los datos se descubrió que muchos elementos pequeños presentes en el terreno se habían clasificado incorrectamente como ruido. En esta categoría se incluyen elementos como farolas, algunos puntos de vegetación y puntos correspondientes a cables, que es el elemento que nos interesa en este trabajo. Se pidió a la empresa adjudicataria del proyecto que solucionara el error y debido a la dificultad de ello entregó los datos de nuevo con una capa por encima del terreno clasificada como ruido con todos los elementos anteriormente comentados.

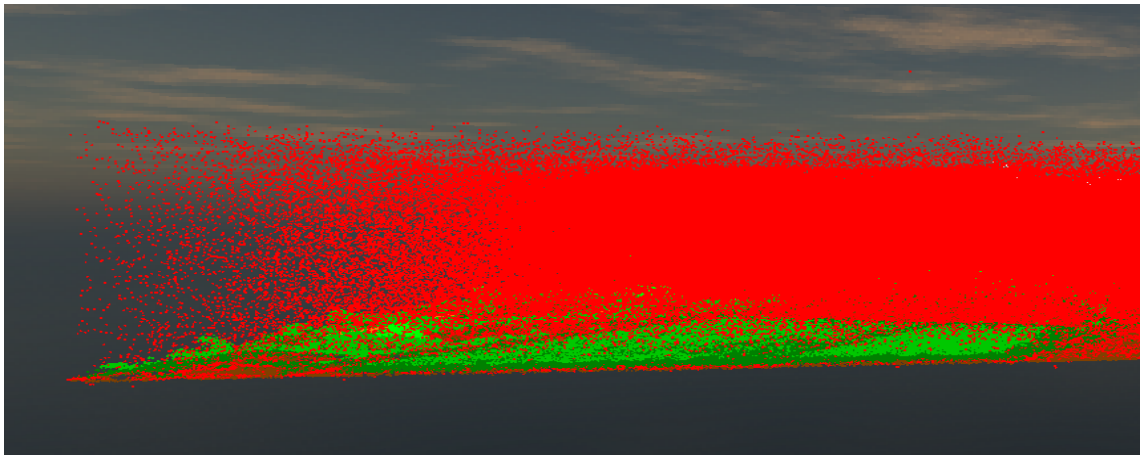


Figura 21: Gran cantidad de ruido (puntos en rojo) en datos LIDAR. Fuente: Elaboración propia.

En un primer momento se pensó utilizar solo esta categoría de ruido para realizar el trabajo y la detección de cables, pero se descubrió posteriormente que muchos de estos cables estaban en esta capa de ruido, aunque había otros muchos incorrectamente clasificados como vegetación alta. Por lo tanto, tras muchas observaciones de los datos se decidió utilizar todos los puntos de cada bloque tanto para la realización del entrenamiento de los algoritmos como para la posterior clasificación.

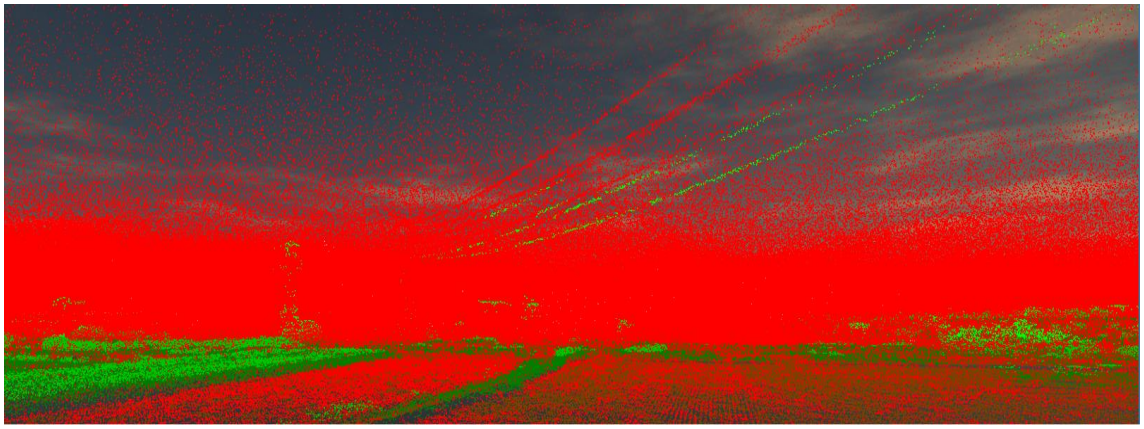


Figura 22: Cable clasificados como ruido (rojo) y como vegetación (verde). Fuente: Elaboración propia.

3.3 Metodología

3.3.1 Introducción

En el estado del arte se han analizado los métodos de clasificación de cables y líneas de alta tensión existentes, la clasificación de datos LiDAR mediante minería de datos y posteriormente una introducción a la teoría sobre minería de datos y los diferentes algoritmos de clasificación que vamos a utilizar. En esta parte del trabajo se propone una metodología que utilice este tipo de algoritmos para realizar la clasificación de líneas de alta tensión en los datos LiDAR utilizados en este proyecto de manera automática.

En este punto hay que tener en cuenta que la minería de datos necesita una gran cantidad de datos para realizar su entrenamiento y así poder predecir una clasificación a aplicar en otro conjunto de datos diferente, obteniendo de manera automática en estos nuevos datos los valores que se les va a predecir en función del algoritmo aplicado (Han & Kamber, 2000). Para conseguir esto deberemos decidir la forma en la que se va a procesar la información y que tipo de información van a contener las áreas de entrenamiento.

3.3.2 Creación áreas de entrenamiento

3.3.2.1 Estudio de variables a utilizar

En la tabla siguiente podemos ver las variables que presenta cada punto de la nube de puntos que forma el archivo LAS. En este trabajo se va a intentar la detección y clasificación de puntos LIDAR solo mediante las variables que presentan los archivos LAS utilizados en este proyecto, por lo tanto, no se va a recurrir a variables obtenidas mediante otros archivos, aunque sí que se generaran variables nuevas a partir de los datos de los archivos LAS.

CAMPO	**DESCRIPCIÓN**
X	Coordenada X
Y	Coordenada Y
Z	Coordenada Z
Intensity	Intensidad del punto láser a la llegada al sensor.
Return_Number	Numero de retorno de ese pulso.
Number_of_Returns	Numero de retornos detectados en ese pulso.
R	Valor asociado al canal Rojo.
G	Valor asociado al canal Verde.
B	Valor asociado al canal Azul.
Classification	Clasificación asignada a ese punto.
Scan_Direction_flag	Dirección del espejo del escáner.
Edge_of_flight_line	Borde de línea de vuelo.
Scan_Angle	Angulo de escaneo.
User_Data	Campo a rellenar por el usuario según sus necesidades.
Point_Sourde_ID	Identificador de pasada.

Tabla 5: Variables presentes en un archivo LAS, formato 1.4. Fuente: ASPRS

Tras observar la tabla anterior se ve que muchas de las variables presenten en el archivo LAS no son útiles para la detección de líneas de alta tensión. Como se ha explicado en la revisión del formato LAS, campos como la dirección del espejo del escáner, el borde de la línea de vuelo, el tiempo de captura o el número de punto no van a aportar ninguna información relevante que pueda hacer que nuestro algoritmo de detección de líneas de alta tensión funcione mejor.

Por lo tanto, para la detección de líneas eléctricas de alta tensión de este trabajo se han decidido utilizar las variables siguientes:

CAMPO	DESCRIPCIÓN
Z	Coordenada Z
Intensity	Intensidad del punto láser a la llegada al sensor.
Return_Number	Numero de retorno de ese pulso.
Number_of_Returns	Numero de retornos detectados en ese pulso.
R	Valor asociado al canal Rojo.
G	Valor asociado al canal Verde.
B	Valor asociado al canal Azul.
Classification	Clasificación asignada a ese punto.

Tabla 6: Variables escogidas para la creación de áreas de entrenamiento. Fuente: Elaboración propia.

En este punto se debe comentar que se ha decidido no utilizar variables externas que hubieran podido ayudar a la detección de cables, tales como variables obtenidas a partir de un MDE, por razones principalmente de coste computacional y de automatización de los procesos de detección de líneas eléctricas.

La creación de una variable de altura del punto respecto al suelo obtenida a partir del MDE de la zona hubiera sido muy interesante de cara a la posterior detección, pero dado que cada variable que se calcule en las áreas de entrenamiento se deberá calcular también en los nuevos datos a clasificar y siendo el objetivo la aplicación el algoritmo a todos los bloques del proyecto sin la necesidad de un proceso previo se ha decidido no hacer uso de archivos ajenos al propio fichero LAS.

De la misma manera se ha pensado en generar variables geométricas entre los puntos y sus puntos vecinos, pero tras múltiples pruebas se ha visto que el coste computacional y, sobre todo, de tiempo necesario, era demasiado elevado para la tarea que se buscaba de poder aplicar el algoritmo una vez entrenado a todos los bloques del proyecto de la zona de estudio. Como se ha expuesto en la parte de presentación cada bloque LAS presenta una media de entre 10 y 20 millones de puntos lo que hace que la generación de variables geométricas entre puntos necesite de un gran tiempo de procesamiento del que no se dispone para el ámbito de este trabajo de investigación.

Además, debemos comentar que se han observado variaciones muy grandes en el valor de intensidad a lo largo del trabajo para el mismo tipo de cubierta de terreno, desconociendo si es debido a la falta de normalización o al uso de múltiples zonas de aplicación, en las que la respuesta espectral para la intensidad puede ser diversa.

Según la literatura consultada los valores de intensidad pueden variar dependiendo de la reflectancia y rugosidad de la superficie, el ángulo de escaneo, el rango de energía considerado, la energía transmitida, los múltiples retornos, la profundidad de la intensidad, el brillo producido por elementos próximos, el tamaño de apertura, la transmisión atmosférica y la humedad (Martínez Blanco, 2016). Debemos comentar que se han encontrado puntos correspondientes a cables con valores muy distintos de intensidad que pueden hacer que no funcione de una manera tan precisa la detección.

En el apartado de resultados se comprobará si es posible la detección de líneas de alta tensión únicamente a partir de los datos del archivo LAS.

3.3.2.2 Extracción de variables

Como se ha expuesto en el punto anterior se van a aprovechar las variables referidas a la coordenada ***Z***, el valor de la intensidad, una nueva variables creada a partir del número de retorno (***r***) y el número de retornos en ese punto (***n***). El valor de la clasificación (***c***), será necesario para la comparación con el resultado obtenido, sin que se haya considerado como variable para el entrenamiento. Además, se utilizarán las variables de color obtenidas de la ortofoto **R**, **G**, **B** tomadas en el mismo momento de captura de los puntos LIDAR.

Por lo tanto las variables serán los valores que presenta cada punto en el archivo LAS excepto la variable de **Retorno** que se calculara dividiendo el número de retorno entre los retornos obtenidos en ese punto, de esta forma el valor de la nueva variable será más útil para la detección de líneas de alta tensión ya que además de ahorrar en coste computacional al necesitar una variable menos, se cree que la información de la variable será más representativa para cada punto que ambas variables por separado.

Por lo tanto, las variables que poseerá cada punto para entrenar al algoritmo serán:

- Coordenada Z
- Intensidad
- Retorno
- R
- G
- B
- Clasificación

El valor de clasificación será necesario para entrenar a los algoritmos de aprendizaje supervisado, se ha decidido darle un valor 1 a todos los puntos que correspondan a cables y un valor 0 al resto de puntos del fichero de entrenamiento que correspondan al resto de opciones (tanto elementos del terreno como edificios vegetación y puntos con ruido).

En este punto se debe comentar que el valor de la variable clasificación no se obtendrá a partir del archivo LAS ya que como hemos comentado en puntos anteriores la información que presenta esta variable en los datos de partida no es correcta en todos los puntos, el valor de esta variable lo asignaremos posteriormente mediante la detección manual de cables en las zonas de entrenamiento. Las nuevas zonas a clasificar tras entrenar a los algoritmos presentaran esta variable sin información y será el algoritmo elegido el encargado de darle un valor 0 si lo clasifica como no cable y 1 si lo clasifica como cable.

	OBJECTID *	X	Y	Z	INTENSIDAD	RED	GREEN	BLUE	Valor_Ret	CLASIFICACION
	1	593501.322	4750848.376	561.744	7136	23296	16640	10496	100	0
	2	593501.152	4750848.386	561.904	7313	23296	16384	11008	100	0
	3	593501.012	4750848.606	561.794	7252	23296	16384	11008	100	0
	4	593501.082	4750848.646	561.784	7237	23296	16384	11008	100	0
	5	593500.912	4750848.996	561.864	7284	23296	15872	10496	100	0
	6	593501.292	4750848.826	561.844	7118	22016	15360	9216	100	0
	7	593501.412	4750848.516	561.864	7031	22528	15872	9728	100	0
	8	593501.972	4750848.646	561.714	6947	22016	15360	9216	100	0
	9	593501.702	4750848.836	561.784	6891	22784	16128	9984	100	0
	10	593501.882	4750848.876	561.764	6937	22784	16128	9984	100	0
	11	593500.212	4750849.556	561.864	7310	22272	15616	9472	100	0
	12	593500.662	4750849.586	561.884	7199	23040	15872	9728	100	0

Tabla 7: Ejemplo de puntos que forman las áreas de entrenamiento. Fuente: Elaboración propia.

Aunque en la tabla anterior se muestra la existencia de las columnas pertenecientes a las coordenadas X e Y estas no se usaran en el entrenamiento de algoritmos, es necesario mantenerlas en el archivo para la posterior representación de los puntos al transformarlos a un archivo LAS.

3.3.2.3 Elección de las áreas de aprendizaje.

En este apartado debemos comentar según la bibliografía consultada que cuantas más áreas de aprendizaje poseamos tanto para entrenar el algoritmo como para testearlo después, mejores serán los resultados de la clasificación (Sanz-Delgado, 2018), por lo tanto, este punto del trabajo será uno de los procesos más importantes a la hora de obtener resultados satisfactorios. Por ello clasificaremos un total de 15 bloques LAS en diferentes partes de la zona de estudio para intentar obtener muestras representativas de cada posible cable que nos encontraremos en la clasificación posterior.

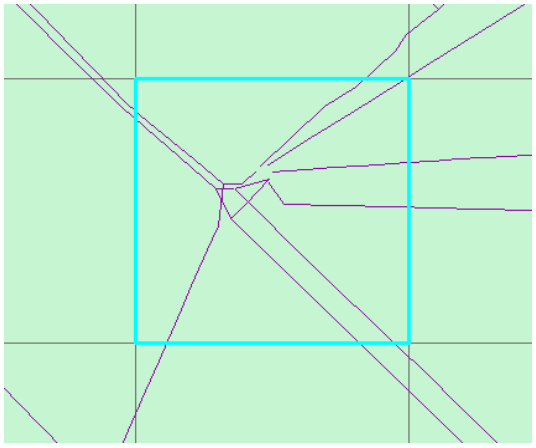


Figura 23: Ejemplo de un bloque LAS de 1km de lado los cuales forman el proyecto LIDAR. Fuente: Elaboración propia.

Para encontrar las zonas donde tenemos cables de alta tensión nos ayudaremos de los mapas vectoriales que actualmente existen de las líneas de alta tensión presentes en el territorio de estudio como se puede ver en la figura 24.

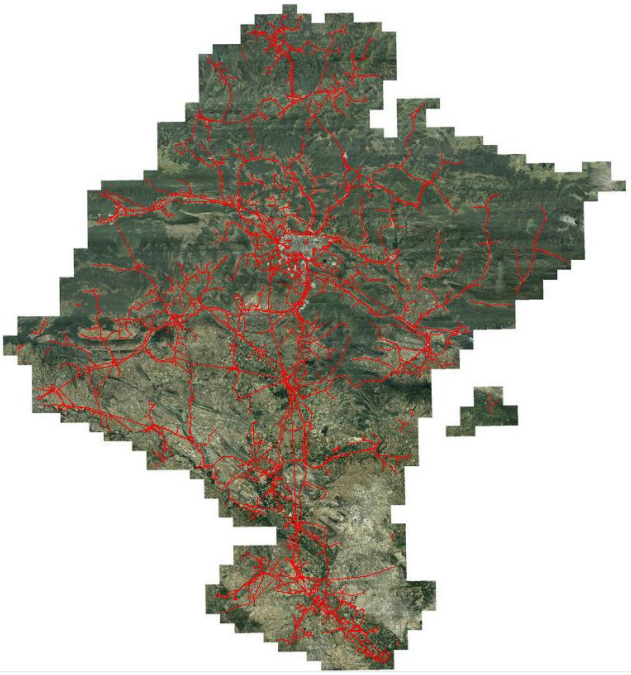


Figura 24: Mapa de líneas eléctricas de Navarra. Fuente: Elaboración propia.

Una vez escogidas las áreas a partir de las cuales se crearán las zonas de aprendizaje será necesario saber a qué bloque de puntos pertenecen, para ello nos valdremos del software ArcGis, una capa con las líneas de alta tensión de la Comunidad Foral de Navarra, una capa con la información de los bloques LAS para saber que bloque necesitaremos abrir para extraer los cables y una ortofoto de todo Navarra de Fondo para saber el tipo de terreno sobre el que esta cada cable y así poder abarcarlos todos. Además, deberemos elegir zonas que presenten más de una pasada de solape y zonas con una única pasada para tener zonas con diferentes densidades de puntos y cubrir todas las posibles opciones.

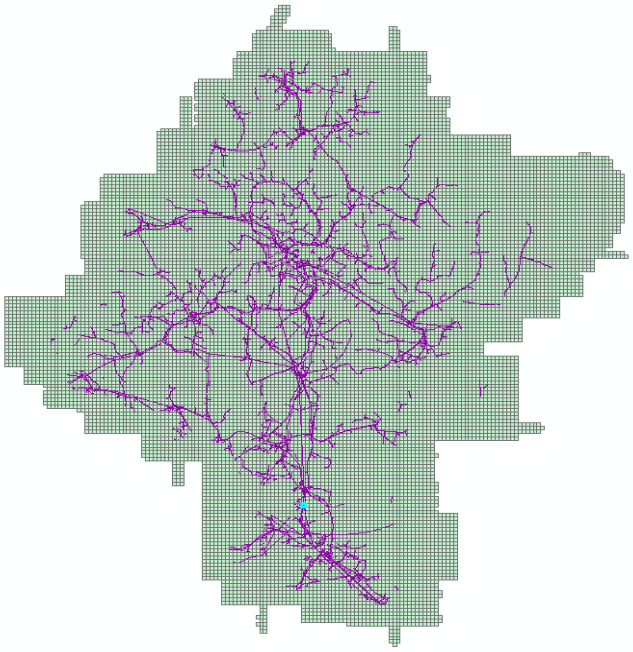


Figura 25: Líneas eléctricas de Navarra y mapa con los bloques que forman el proyecto. Fuente Elaboración propia.

Con los mapas anteriores superpuestos se ha tratado de seleccionar todas las casuísticas posibles en las que podemos encontrarnos cables en todo el territorio de estudio. Se han seleccionado bloques que presenten cables en zonas de bosque, zonas de cultivos, zonas montañosas y zonas urbanas. Además, como se ha comentado se han seleccionado bloques que presentan líneas de alta tensión en zonas con una pasada de vuelo y en zonas con múltiples pasadas. Las zonas del proyecto de donde se han obtenido los bloques se muestran en la figura 26.

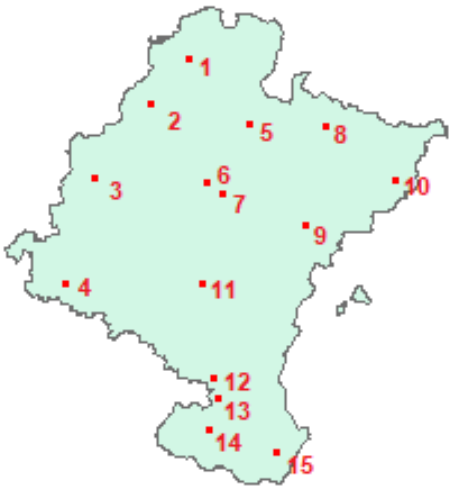


Figura 26: Mapa con las zonas donde se han escogido bloques para formas las áreas de entrenamiento. Fuente: Elaboración propia.

En la figura 27 se pueden ver algunos de los bloques anteriormente seleccionados para la creación de las áreas de aprendizaje. En estas imágenes se ha querido incluir un ejemplo de cada una de las casuísticas mencionadas anteriormente. Como se puede observar también se han escogido zonas con un gran número de líneas de alta tensión y zonas con un menor número de líneas para intentar cubrir todas las posibilidades.

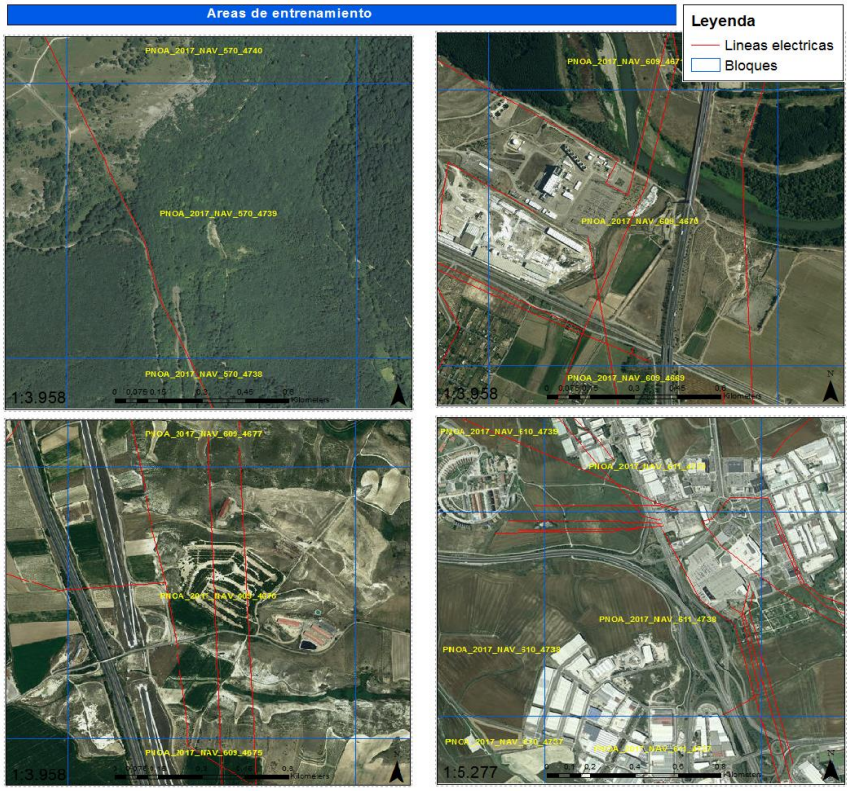


Figura 27: Ejemplo de bloques que componen las áreas de entrenamiento. Fuente: Elaboración propia.

En la tabla siguiente se pueden observar los números de cada bloque y el tipo de cubierta.

Zona	Nombre del Bloque	Ámbito
1	PNOA_2017_NAV_588_4786	Forestal
2	PNOA_2017_NAV_586_4773	Forestal
3	PNOA_2017_NAV_564_4748	Forestal
4	PNOA_2017_NAV_553_4708	Cultivos
5	PNOA_2017_NAV_626_4758	Forestal
6	PNOA_2017_NAV_607_4745	Urbano
7	PNOA_2017_NAV_612_4738	Urbano
8	PNOA_2017_NAV_644_4762	Forestal
9	PNOA_2017_NAV_649_4731	Urbano
10	PNOA_2017_NAV_666_4739	Forestal
11	PNOA_2017_NAV_603_4693	Cultivos
12	PNOA_2017_NAV_610_4674	Cultivos
13	PNOA_2017_NAV_608_4670	Urbano
14	PNOA_2017_NAV_601_4663	Urbano
15	PNOA_2017_NAV_628_4645	Cultivos

Tabla 8: Zonas escogidas como áreas de aprendizaje: Fuente: Elaboración propia.

3.3.2.4 Extracción de líneas de alta tensión

Una vez escogidas las zonas de aprendizaje se procede a la búsqueda y extracción de los cables presentes en ellas. En este trabajo se han seleccionado un total de 15 bloques de datos del vuelo LiDAR presentando muchos de esos bloques más de una línea de alta tensión diferente. Generalmente se han seleccionado zonas que presenten una gran cantidad de cables para extraer más puntos que correspondan a cables. El proceso de extracción de cada cable ha sido quizás la parte más costosa del trabajo ya que antes de empezar el trabajo no se contaba con ningún cable clasificado y tras probar con la mayoría de programas específicos para ello todos arrojaban resultados poco satisfactorios.

Para la detección y extracción de las líneas de alta tensión se va a utilizar el software TerraScan y más concretamente el modulo que posee para datos LIDAR llamado *TerraSolid*. Este módulo presenta una herramienta específica para la detección de cables que se explicara a continuación. La herramienta se llama *Detect Wires* (Detectar cables) y se utiliza para vectorizar cables y clasificar puntos en los cables de una línea eléctrica. Esta herramienta dibuja en el archivo donde estamos trabajando todos los cables detectados en formato vectorial. Para utilizarla los datos deben estar cargados en la ventana de TerraScan (Soininen, 2016).

La herramienta busca puntos a lo largo de una curva de catenaria. Las curvas de catenaria son descripciones matemáticas de cables que están conectados en sus puntos extremos pero que cuelgan libremente entre estos puntos extremos. El proceso involucra los mínimos cuadrados que encajan tanto para la ecuación de la recta xy como para la ecuación de la curva de elevación de la variable catenaria (Soininen, 2016).

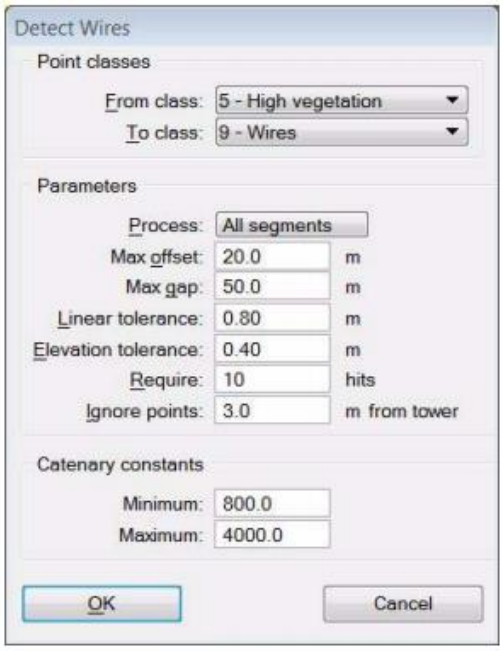


Figura 28: Herramienta para detectar cables en software TerraSolid. Fuente: Terrascan

El parámetro más importante que controla la detección de cables es la separación máxima (Max gap), que define el máximo entre puntos láser consecutivos en un cable. No es aconsejable ejecutar la detección en su totalidad con un gran valor de gap máximo, ya que aumenta la posibilidad de que se produzcan falsas detecciones. Se recomienda realizar la detección primero con un valor de separación máximo relativamente pequeño con el que no se detecten todos los

cables. Para ubicaciones con menos puntos en los cables, la detección debe ser para segmentos individuales con un valor menor (Soininen, 2016).

Parámetro	**Efecto**
Clase de partida	Clase de punto a partir del cual se detectan los cables
Clase final	Clase de objetivo en la que se clasifican los puntos de los cables.
Proceso	Determina dónde se detectan los cables: - Todos los segmentos - para todos los segmentos para los que se cargan datos en TerraScan. - Segmento único - sólo para la torre seleccionada.
Desviación máxima	Distancia máxima desde la línea que forman las torres tanto a izquierda como a derecha. Define el espacio en el que el software busca cables.
Distancia Máxima	Máximo espacio permitido entre puntos consecutivos en un cable
Tolerancia lineal	Tolerancia para el ajuste de la línea xy de clasificación de puntos en un cable.
Tolerancia vertical	Tolerancia para el ajuste de la curva de elevación y clasificación de los puntos en un cable.
Requerimientos mínimos	Se requiere una cantidad mínima de puntos láser en un solo cable para detección. Los valores pueden variar de 3 a 999.
Ignorar puntos	Distancia desde la torre dentro de la cual los puntos son ignorados para la detección de cables. Los puntos cercanos a la torre pueden ser de estructuras de torre y debe ser ignorado cuando se determina el valor matemático de forma del cable.
Mínimo	Constante de catenaria mínima para aceptar un cable.
Máximo	Constante de catenaria máxima para aceptar un cable.

El resto de parámetros necesarios para la detección de cables en este software son los siguientes:

Tabla 9: Parámetros necesarios para la detección de cables. Fuente: Terrascan.

Variando estos parámetros es posible detectar y clasificar líneas de alta tensión en los bloques escogidos. En esta parte del trabajo se debe comentar que, aunque parezca sencillo detectar los cables con esta herramienta los resultados no son fáciles de obtener. En una primera aproximación se intentó utilizar la herramienta en todo el bloque LAS elegido, pero se descubrió que el tiempo de procesamiento era muy elevado y el porcentaje de acierto era bastante bajo. Variando los parámetros se consiguió llegar a detectar algunas de las líneas de alta tensión en alguno de los bloques, pero se descubrió que los parámetros que encontraban cables en un bloque no tenían por qué encontrar en otros bloques.

Además, se vio que una variación pequeña en alguno de los parámetros conllevaba muchos cambios en la detección de cables. Debido al tiempo de procesamiento elevado que necesitaba cada vez que se ejecutaba el algoritmo de detección de cables se decidió seleccionar una zona alrededor de cada cable en cada bloque en el que el algoritmo de detección de cables actuara para reducir el área de búsqueda. Con esto se redujo considerablemente el tiempo de procesamiento de cada zona y se pudieron realizar más intentos de detección y mejorar los resultados.

Una vez seleccionadas las zonas y variando los parámetros del detector se consiguió llegar a una detección aceptable de líneas de alta tensión. Cabe comentar en esta punto que debido a la gran cantidad de ruido que presentan los datos LIDAR de este proyecto si se selecciona un valor de distancia máxima demasiado pequeña el algoritmo detecta cables casi entre todos los puntos de ruido del bloque y si este parámetro es demasiado grande el algoritmo no encuentra ningún cable. Por tanto, aunque parezca que la detección es sencilla ha supuesto gran cantidad del tiempo invertido en el proyecto. Además, como se ha comentado los parámetros que funcionan en un bloque no arrojan los mismos resultados en otros bloques. En la siguiente imagen podemos ver una línea de alta tensión del proyecto extraída con el procedimiento expuesto anteriormente. Como se puede apreciar incluso en el mejor de los casos se observan algunos puntos incorrectamente clasificados, en este caso como vegetación. Para la creación de las áreas de entrenamiento posteriores hubo que seleccionar y clasificar manualmente estos puntos lo que también aumentó el tiempo de realización del estudio considerablemente.

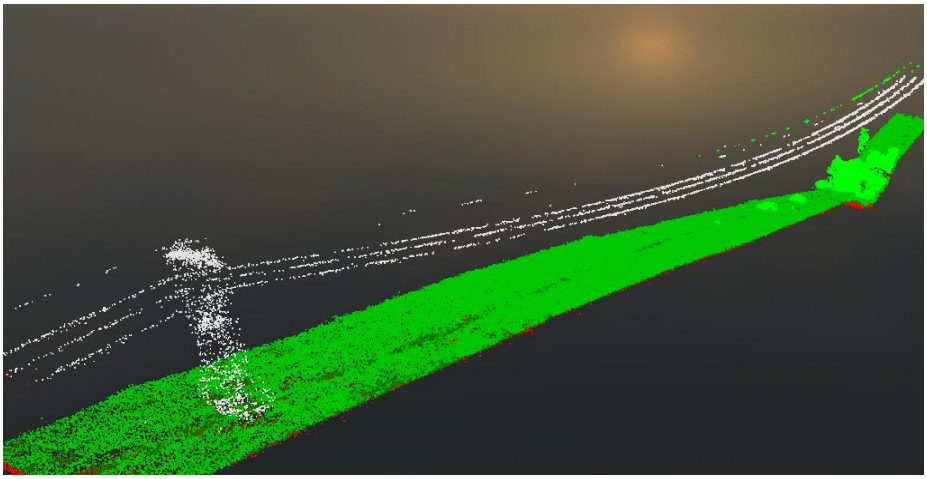


Figura 29: Cable extraído y clasificado con el software TerraSolid. Fuente: Elaboración propia

3.3.2.5. Creación del fichero de aprendizaje.

Una vez que todos los cables de los bloques seleccionados se han clasificado el siguiente paso es la creación de un CSV con los ejemplos de aprendizaje. Para intentar hacer un primer balanceo de los datos para que el número de puntos de cable y no cable sea similar se selecciona una zona alrededor de cada cable y se exporta a un fichero LAS conjunto a partir del cual se creará un CSV con los ejemplos de aprendizaje. Con esta medida se reduce el número de puntos de no cable en relación al número de puntos de cable. Aunque esto reduzca el número de puntos de no cable tras la realización de los primeros ensayos de descubrió que es necesario realizar un proceso de remuestreo o balanceo del conjunto de aprendizaje mediante algoritmos. Este procedimiento se explicará en posteriores apartados.

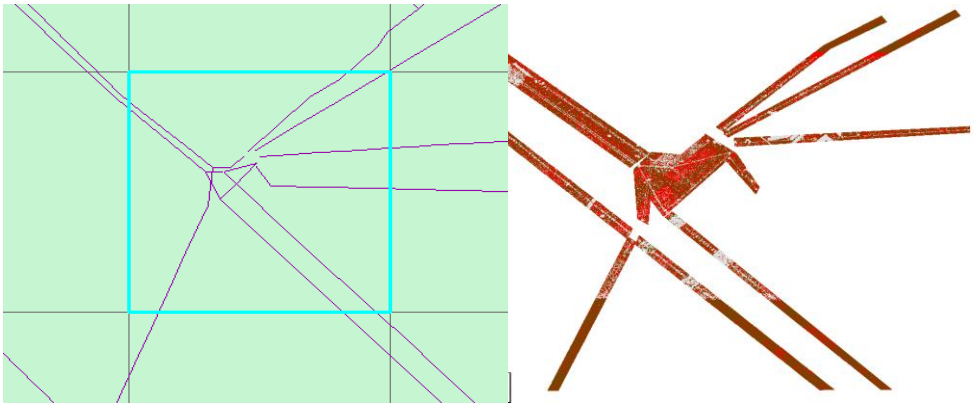


Figura 30: Selección de zonas próximas a los cables para balancear los datos. Fuente: Elaboración propia.

Con todas las zonas de aprendizaje en un único fichero nos ayudaremos del software ArcGis para la elección de las variables que formaran parte de dicho fichero y para la creación de la variable referente al valor del retorno del punto a partir del número de retorno y del número de retornos en de ese punto. Finalmente, con la librería *LASTools* y concretamente el comando *LAStotxt* conseguiremos un fichero CSV con la información de todos los puntos pertenecientes a las áreas de aprendizaje y sus variables asociadas con el que pasaremos al balanceo de los datos para el posterior entrenamiento de los algoritmos.

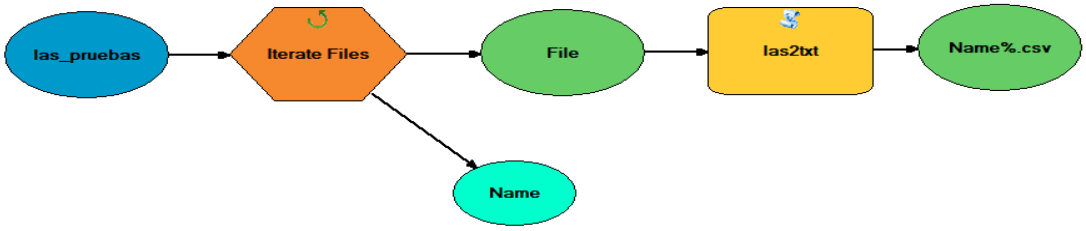


Figura 31: Modelo creado para la creación de las áreas de entrenamiento. Fuente: Elaboración propia.

3.3.2.6 División del conjunto de aprendizaje

Antes de realizar el balanceo de los datos mediante algoritmos en primer lugar se deberá realizar la partición de las áreas de aprendizaje en dos conjuntos; entrenamiento y test. La partición del conjunto de aprendizaje se realizará mediante el método hold-out. Este método divide los datos de aprendizaje aleatoriamente en dos conjuntos. Estos conjuntos de datos son mutuamente excluyentes, cada punto que forma los datos de aprendizaje solo puede estar en uno de los dos conjuntos. El conjunto de entrenamiento es usado para entrenar el algoritmo seleccionado, posteriormente se utiliza el conjunto de test para estimar el rendimiento del algoritmo (Robles Porcada, 2003).

Generalmente el conjunto de datos utilizado para entrenamiento debe ser mayor que el conjunto de datos de test (Sanz-Delgado, 2018*)*. En este trabajo se utilizará un 75% de los datos como conjunto de entrenamiento y el 25% restante como test.

La llamada a dicha función es la siguiente:

X_train, X_test, y_train, y_test = *model_selection.train_test_split(inputData, outputData, train_size=porcentajeTrain, random_state=semilla)*

- Los parámetros de entrada son:

  - **inputData**: los datos de entrada, el campo data del objeto dataset.
  - **outputData**: los datos de salida, el campo target del objeto dataset.
  - **porcentajeTrain**: la proporción de ejemplos de entrenamiento (entre 0 y 1).
  - **semilla**: valor que determina la semilla para la generación de números aleatorios.
- Los parámetros de salida son:
  - **X_train**: los datos de entrada del conjunto de entrenamiento.
  - **X_test**: los datos de entrada del conjunto de test.
  - **y_train**: los datos de salida del conjunto de entrenamiento.
  - **y_test**: los datos de salida del conjunto de test.

3.3.2.7 Balanceo de áreas de entrenamiento

En este apartado del trabajo se va a realizar un remuestreo o balanceo de los datos de entrenamiento. Como hemos explicado en la parte de antecedentes del trabajo los algoritmos de aprendizaje supervisado funcionan mejor con conjuntos de datos balanceados, esto quiere decir, que tiene que existir aproximadamente el mismo número de datos de cada clase. En este caso concreto deberían existir el mismo número de puntos correspondientes a cable y no cable. Para realizar esto se aplican métodos de remuestreo a los datos de entrenamiento para balancear el conjunto de datos de entrenamiento. Estos métodos están basados en dos técnicas que se explicarán a continuación:

- **Métodos de Under-Sampling**, estos métodos eliminan objetos de la clase mayoritaria, en nuestro caso los ejemplos de no cable, con el objetivo de crear un equilibrado conjunto de datos. El principal inconveniente del enfoque de estos métodos es que puede excluir algunos objetos representativos del conjunto de entrenamiento afectando de esta manera el modelo construido por el clasificador (Jes, 2016).

- **Métodos de Over-Sampling**, la idea principal es la creación de nuevos objetos de la clase minoritaria, en nuestro caso los ejemplos de cable, para producir unos nuevos conjuntos de datos con una distribución equilibrada de clase. Sin embargo, el principal inconveniente del enfoque de sobre muestreo es que puede incluir también muchos objetos artificiales que pueden producir sobreajuste (Jes, 2016).

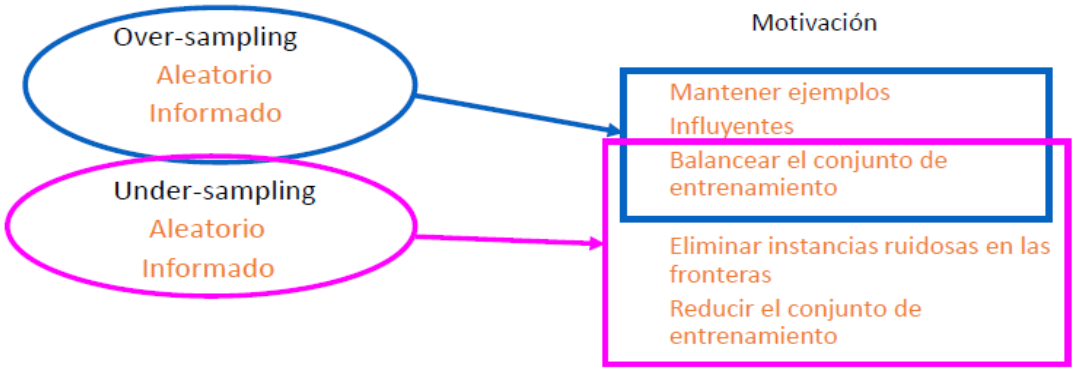


Figura 32: Características y motivación de cada método de balanceo de datos. Fuente: Sanz-Delgado

En este trabajo se van a utilizar los métodos que se describen a continuación y posteriormente se escogerá el que obtenga mejores resultados:

Métodos de under-sampling

RUS (Random under-sampling) Este método se basa en un algoritmo que selecciona de una manera aleatoria instancias de la clase mayoritaria para ser eliminadas sin remplazamiento hasta que ambas clases queden balanceadas (Visa & Ralescu, 2003).

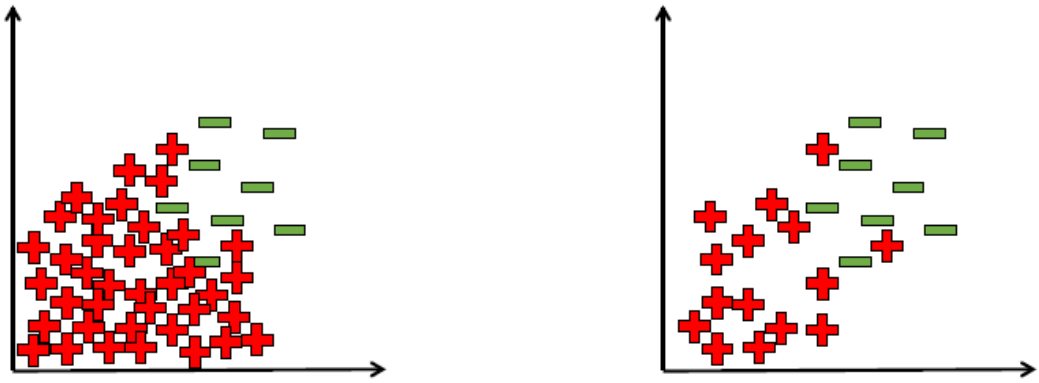


Figura 33: Ejemplo de proceso de balanceo RUS. Fuente: Sanz-Delgado

El principal problema que puede aparecer con este método es que puede eliminar ejemplos potencialmente útiles para el aprendizaje (Sanz, 2018)

CNN (Condensed Nearest Neighbor) se basa en la regla del vecino más cercano (NN). Su propósito es reducir el tamaño de los conjuntos de datos originales mediante la eliminación de ciertos objetos sin afectar el rendimiento del clasificador (Holm, 1979).

El problema que puede suponer aplicar este método es que no se obtenga un conjunto consistente mínimo. Además, este método es sensible al ruido, por lo tanto, con este método los ejemplos ruido serán fallados, lo que puede afectar al rendimiento posterior (Sanz-Delgado, 2018).

TL (Tomek`s modification of Condensed Nearest Neighbor) Un enlace Tomek está compuesto por una pareja de ejemplos de clase diferente que no tengan ningún ejemplo entre ellos. Es decir, que sean su vecino más cercano respectivamente. Cuando tenemos un enlace Tomek uno de ellos es ruido o los dos objetos están en el límite. Entonces antes de la aplicación de la regla del vecino más cercano (NN), este método obtiene un conjunto de objetos que contiene sólo los objetos cerca de los límites de decisión (Jes, 2016). En la figura 34 se puede ver su funcionamiento.

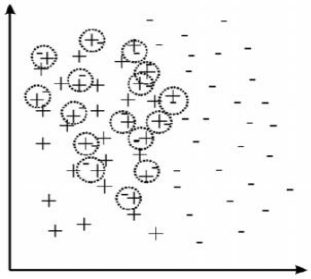


Figura 34: Ejemplo de funcionamiento de método Tomek Link. Fuente: Sanz-Delgado

OSS (One Sided Selection) Aplica secuencialmente CNN y Tomek links. CNN elimina ejemplos de la clase negativa alejados de la frontera de decisión. Tomek link elimina ejemplos de la clase negativa considerados como ruido o ejemplos de la frontera (Sanz-Delgado, 2018).

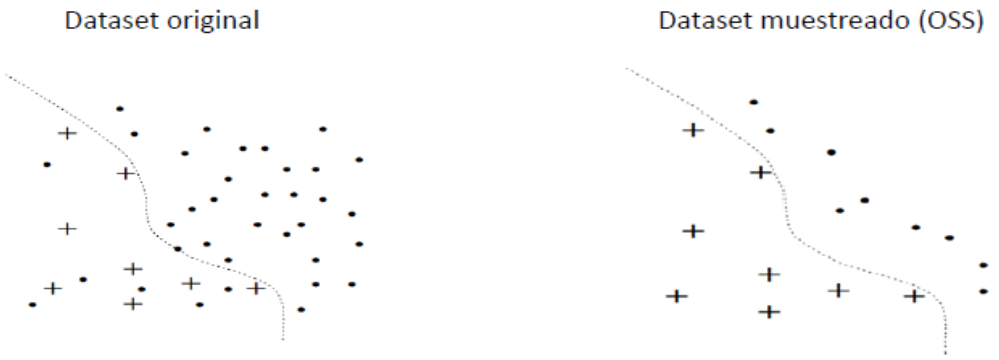


Figura 35: Esquema de trabajo de método OSS. Fuente: Sanz-Delgado

NCL-I (Neighborhood Cleaning Rule) Este algoritmo usa la regla ENN para eliminar objetos de la clase mayoritaria. ENN elimina cualquier objeto cuya clase de etiqueta difiere de la clase de al menos tres de sus cinco vecinos más cercanos (Jes, 2016).

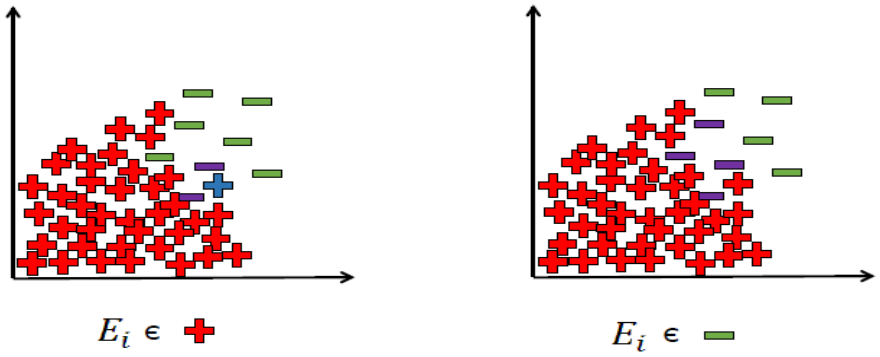


Figura 36: Esquema de trabajo de método NCL. Fuente: Sanz-Delgado

Métodos de over-sampling

ROS-I (Random over-sampling) Este método replica objetos aleatoriamente en la clase minoritaria hasta que ambas clases tienen el mismo número de objetos. (Sanz-Delgado, 2018)

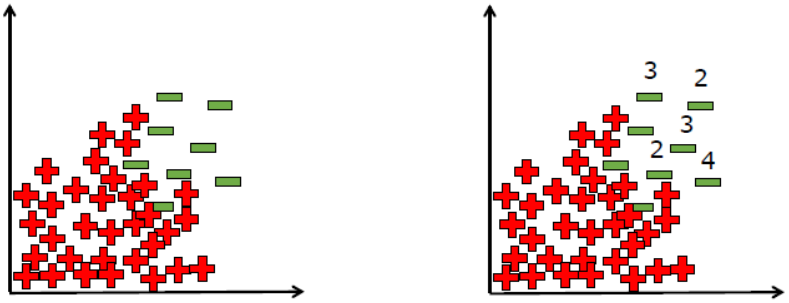


Figura 37: Esquema de trabajo de metodo ROS. Fuente: Sanz-Delgado

SMOTE-I (Synthetic Minority Over-sampling Technique) Este algoritmo para cada ejemplo de la clase minoritaria introduce ejemplos sintéticos en el híper plano que forma el elemento con uno de sus K vecinos más cercanos. Los nuevos objetos se generan por medio de interpolación entre el objeto y uno de sus vecinos más cercanos. (Sanz-Delgado, 2018). El funcionamiento de este método se muestra en la figura 38.

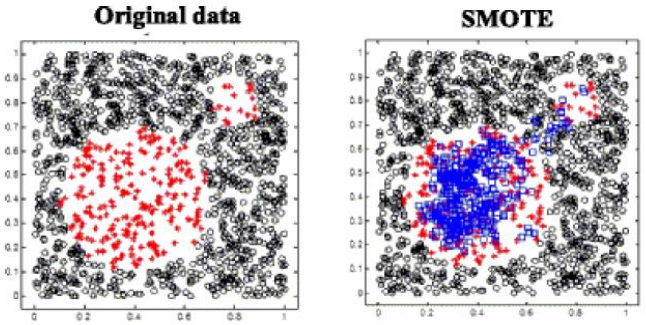


Figura 38: Ejemplo de aplicación del método SMOTE. Fuente: Sanz-Delgado.

Este método obliga a la región de decisión de la clase minoritaria a ser más general. Sin embargo, este método presenta el problema de que puede introducir ejemplos de la clase minoritaria en el área de la mayoritaria, es decir crea malos ejemplos que pueden confundir a los clasificadores. (Sanz-Delgado, 2018)

SMOTE-ENN-I (Synthetic Minority Over-sampling TEchnique + Edited Nearest Neighbor) Este es un método híbrido que aplica SMOTE y a continuación ENN, explicado anteriormente (Visa & Ralescu, 2003).

SMOTE-TL-I (Synthetic Minority Over-sampling TEchnique + Tomek's modification of Condensed Nearest Neighbor) Este un método es un híbrido que utiliza SMOTE y después TOMEK LINK para eliminar en este caso ejemplos ruidosos o ejemplos en la frontera. En este caso se eliminan los ejemplos de las dos clases no solo la mayoritaria. (Visa & Ralescu, 2003).

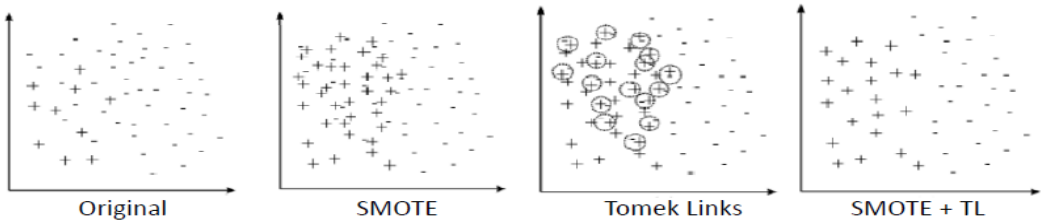


Figura 39: Esquema de trabajo de método hibrido SMOTE + Tomek Link. Fuente: Sanz-Delgado

Como hemos explicado anteriormente estos algoritmos de balanceo se aplican al conjunto de datos de la muestra de aprendizaje destinados a entrenamiento ya que el conjunto de datos que se utilice como test no hay que balancearlo. Por lo tanto, primero se divide el conjunto de datos de aprendizaje con los métodos explicados anteriormente y con el resultado que se puede ver en la tabla 10.

Ejemplos totales	**4.527.722**
Ejemplos entrenamiento	3.395.791
Ejemplos test	1.131.931

Tabla 10: Composición de los ejemplos de aprendizaje. Fuente: Elaboración propia.

Una vez dividido el conjunto de aprendizaje en entrenamiento y test procedemos a aplicar los métodos de balanceo. Los ejemplos de cada clase en el conjunto de datos de aprendizaje original podemos verlo en la tabla 11.

Conjunto de entrenamiento	**3.395.791**
Ejemplos cable	69.078
Ejemplos no cable	3.326.713

Tabla 11: Ejemplos de cada clase en el conjunto de entrenamiento original. Fuente: Elaboración propia.

Los resultados obtenidos tras realizar los métodos de balanceo podemos verlos en la tabla 12.

METODOS UNDER SAMPLING		METODOS OVER SAMPLING	
RUS - Random Under Sampling		ROS - Random Over Sampling	
Ejemplos cable	69.078	Ejemplos cable	3.326.713
Ejemplos no cable	69.078	Ejemplos no cable	3.326.713
Ejemplos eliminados	3.257.635	Ejemplos añadidos	3.257.635
TL - Tomek Link		SMOTE	
Ejemplos cable	69.078	Ejemplos cable	3.326.713
Ejemplos no cable	3.315.624	Ejemplos no cable	3.326.713
Ejemplos eliminados	11.089	Ejemplos añadidos	3.257.635
CNN - Condensed Neighbour		SMOTE + Tomek Link	
Ejemplos cable	69.078	Ejemplos cable	3.326.713
Ejemplos no cable	3.316.634	Ejemplos no cable	3.320.383
Ejemplos eliminados	10.079	Ejemplos añadidos	3.251.305
OSS - One Sise Selection		SMOTE + ENN	
Ejemplos cable	69.078	Ejemplos cable	3.326.713
Ejemplos no cable	3.315.564	Ejemplos no cable	3.267.461
Ejemplos eliminados	11.153	Ejemplos añadidos	3.198.383
NCL - Neighbourhood Cleaning Rule			
Ejemplos cable	69.078		
Ejemplos no cable	3.266.508		
Ejemplos eliminados	60.205		

Tabla 12: Resultados de los diferentes métodos de balanceo de las áreas de entrenamiento. Fuente: Elaboración propia

Tras observar los resultados obtenidos con cada técnica de remuestreo en las áreas de entrenamiento se ha decidido utilizar una técnica hibrida de remuestreo, realizando primero SMOTE para balancear el conjunto de entrenamiento y posteriormente ejecutando sobre este conjunto TOMEK LINK para eliminar los puntos que formen enlaces TOMEK.

Se ha tomado esta decisión ya que se ha observado que utilizando técnicas de *Under sampling* se eliminan un número excesivo de puntos de entrenamiento lo que hace que los resultados obtenidos al evaluar los algoritmos tras el entrenamiento no sean tan satisfactorios como los resultados obtenidos con las técnicas de *Over-sampling*. Este hecho tiene su justificación ya que si se observa el conjunto de entrenamiento antes de muestrear vemos que el número de puntos de no cable es unas 45 veces superior al conjunto de números de cables por lo tanto se cree que

eliminar tantos puntos resulta perjudicial para el entrenamiento posterior ya que estaremos eliminando puntos importantes para realizar el aprendizaje.

3.3.3 Aplicación de algoritmos de aprendizaje automático

3.3.3.1 Parámetros utilizados en cada algoritmo de clasificación.

En la parte de antecedentes del presente trabajo se ha explicado la parte teórica de los algoritmos de clasificación que se utilizaran en el presente trabajo. Estos algoritmos junto con las explicaciones necesarias para su utilización están disponibles en la librería scikit-learn (Scikit-learn, 2017). En esta parte del trabajo se expondrán los parámetros utilizados en cada algoritmo de clasificación.

1. **Arboles de decisión**

   En este trabajo de investigación se utilizan dos de los algoritmos más utilizados para generar árboles de decisión: C4.5 y CART. Estos algoritmos se pueden utilizar para determinar el árbol de decisión a utilizar (C4.5 o CART) así como varios de sus parámetros que determinarán su posterior comportamiento.

   La llamada al constructor del árbol de decisión es la siguiente:

   clasificador = *tree.DecisionTreeClassifier(criterion=tipoMedidaImpurezaNodo, min_samples_split=numeroMinimoEjemplosRama, min_samples_leaf= numeroMinimoEjemplosHoja)*

   Los valores de los parámetros que determinan el comportamiento del clasificador son los siguientes:

   - ***criterion***: tipo de medida de la impureza del nodo.
     - 'gini': indice GINI, árbol de decisión CART.
     - 'entropy': ratio de ganancia de información, árbol de decisión C4.5.
   - ***min_samples_split***: número mínimo de ejemplos necesario para dividir un nodo (por defecto es 2).
   - ***min_samples_leaf***: número mínimo de ejemplos para generar una hoja (por defecto es 1).

   En este trabajo se utilizarán tanto el tipo de impureza del nodo 'gini' (*CART*) y 'entropy' (*C4.5*). Los números máximos y mínimos de hojas se dejarán por defecto.

2. **Métodos basados en Multi-clasificadores (*Ensembles*)**

   En este trabajo vamos a utilizar los árboles de decisión como clasificador base para generar los diferentes ensembles basados en variaciones de datos ya que es válido para todos ellos: Bagging, Adaboost y Random Forest.

   2.1 Adaboost.

   La llamada al constructor y sus principales parámetros de entrada son los siguientes:

   AdaBoostClassifier*=(base_estimator=clasificadorBase,n_estimators=numeroClasificadoresBase, random_state=semilla)*

Los parámetros de entrada son:

- ***base_estimator***: objeto de una clase correspondiente a un clasificador. Por defecto se utilizan los árboles de decisión.
- ***n_estimators***: valor entero que determina el número de clasificadores que compondrán el ensemble. Por defecto es 50.
- ***random_state***: valor que determina la semilla para la generación de números aleatorios.

En este trabajo se utilizará Adaboost con 20 clasificadores base los cuales serán árboles de decisión C4.5. Además, se utilizará el valor 12 como semilla.

2.2 Bagging.

La llamada al constructor y sus principales parámetros de entrada son los siguientes:

BaggingClassifier*=(base_estimator=clasificadorBase,n_estimators=numeroClasificadoresBase, random_state=semilla)*

Los parámetros de entrada son:

- ***base_estimator***: objeto de una clase correspondiente a un clasificador. Por defecto se utilizan los árboles de decisión.
- ***n_estimators***: valor entero que determina el número de clasificadores que compondrán el ensemble. Por defecto es 50.
- ***random_state***: valor que determina la semilla para la generación de números aleatorios.

En este trabajo se utilizará Bagging con 20 clasificadores base los cuales serán árboles de decisión C4.5. Además, se utilizará el valor 12 como semilla.

2.3 Random Forest.

La llamada al constructor y sus principales parámetros de entrada son los siguientes:

RandomForestClassifier*(criterion=medidaImpureza, max_features = numVariables, n_estimators=numeroClasificadoresBase, random_state=semilla)*

Los parámetros de entrada son:

- ***criterion***: tipo de medida de impureza ('*gi*ni' o *'entropy*'). Por defecto es gini.
- ***max_features***: número de variables a evaluar en cada nodo de cada árbol de decisión que forma en *Random forest*. Los valores que puede tomar son:
  - '*auto*': hace la raíz cuadrada del número de variables (valor por defecto).
  - '*sqrt*': hace la raíz cuadrada del número de variables.
  - '*log2*': hace el logaritmo en base dos del número de variables.
- ***n_estimators***: valor entero que determina el número de clasificadores que compondrán el *ensembl*e. Por defecto es 10.
- ***random_state***: valor que determina la semilla para la generación de números aleatorios.

En este trabajo se utilizará *Random Forest* con 20 árboles de decisión C4.5, utilizando la raíz cuadrada como número de variables a comprobar en el árbol de decisión. Además, se utilizará el valor 12 como semilla.

3. Algoritmos del tipo vecino más cercano (*Nearest Neighbors*)

Estos algoritmos tienen varios parámetros que determinan el comportamiento del algoritmo. La llamada al constructor y sus parámetros son los siguientes:

clasificador = *neighbors.KNeighborsClassifier(n_neighbors = K, weights = tipoVoto, metric = tipoDistancia, p = r)*

Los diferentes parámetros de entrada son:

- ***n_neighbors*** = K: número de vecinos a considerar (valor por defecto = 5)
- ***weights*** = tipoVoto: forma de votar (peso de cada ejemplo cercano). tipoVoto puede tomar los siguientes valores:
  - '*uniform*': voto por mayoría (valor por defecto)
  - '*distance*': voto en función de la inversa de la distancia
- ***metric*** = tipoDistancia: forma de calcular la distancia entre los ejemplos. tipoDistancia puede tomar los siguientes valores:
  - '*manhattan*': distancia de manhattan
  - '*euclidean*': distancia euclidea
  - '*minkowski*': distancia de Minkowski (valor por defecto)
- ***r***: en caso de utilizar la distancia de Minkowski hay que especificar el valor del parámetro p que se corresponde al exponente r. r puede ser cualquier valor, entre ellos:
  - r = 1: distancia de manhattan
  - r = 2: distancia euclidea (valor por defecto)

En este trabajo se dejarán todos los valores por defecto.

3.3.3.2 Métricas de rendimiento

Una vez entrenado cada algoritmo se obtendrán las métricas de rendimiento del conjunto de test de cada algoritmo. Estas métricas pueden obtenerse ya que el conjunto de datos de test posee un campo con el valor de la clasificación real (cable = 1 o no cable = 0), comparando los datos de la información de partida de las áreas de entrenamiento con los datos predichos por cada algoritmo en el conjunto de test podremos saber el porcentaje de puntos bien clasificados de cada algoritmo.

Para valorar los resultados de cada algoritmo en la fase de entrenamiento vamos a calcular diferentes medidas del rendimiento. Para ello se necesitara la matriz confusión de cada algoritmo. Una matriz de confusión permite ver mediante una tabla, la distribución de los errores cometidos por un clasificador a lo largo de las distintas categorías del problema. En dicha matriz se cruza la clase predicha por el clasificador con la clase real. Uno de los beneficios de las matrices de confusión es que facilitan la visión de si el sistema está confundiendo las clases (Robles Porcada, 2003).

		Clasificación como	
		Si	No
Clase real	SI	Verdadero positivo (VP)	Falso negativo (FN)
	NO	Falso Positivo (FP)	Verdadero Negativo (VN)

Figura 40: Matriz de confusión. Fuente: Sanz-delgado.

Si en los datos de entrada el número de muestras de clases diferentes cambia mucho la tasa de error del clasificador no es representativa de lo bien que realiza la tarea el clasificador. Si por ejemplo hay 990 muestras de la clase 0 y sólo 10 de la clase 1, el clasificador puede tener fácilmente un sesgo hacia la clase 0. Si el clasificador clasifica todas las muestras como clase 0 su ratio de ejemplos clasificados correctamente será del 99%. En este caso puede ocurrirnos esto ya que el número de puntos correspondientes a la clase no cable es mucha mayor que el número de pixeles de la clase cable. Para solucionar este punto además del ***Accuracy*** o ratio de elementos clasificados correctamente del clasificador calcularemos otros parámetros como son los siguientes.

Accuracy

$$Acc = \frac{VP + VN}{VP + VN + FP + FN}$$

Figura 41: Ecuación Accuracy. Fuente: Sanz-Delgado.

Recall: Ejemplos de la clase positiva (clase con pocos ejemplos, clase de interés) clasificados correctamente.

$$TPR = Recall = \frac{VP}{VP + FN}$$

Figura 42: Ecuación Recall. Fuente: Sanz-Delgado.

Precisión: Proporción de ejemplos clasificados en la clase positiva que son realmente de la clase positiva.

$$Precision = \frac{VP}{VP+FP}$$

Figura 43: Ecuación Precisión. Fuente: Sanz-Delgado.

Especificidad: Ejemplos de la clase negativa clasificados correctamente.

$$TNR = Especificidad = \frac{VN}{VN + FP}$$

Figura 44: Ecuación Especificidad. Fuente: Sanz-Delgado.

Media geométrica: Métrica que permite obtener un balance entre los porcentajes de ejemplos bien clasificados de las dos clases, para ello se aplica la media geométrica entre *recall* y especificidad.

Media geométrica

$$GM = \sqrt{TPR \times TNR}$$

Figura 45: Ecuación media Geométrica. Fuente: Sanz-Delgado.

3.3.3.3 Clasificación de nuevas zonas

Una vez entrenados todos los algoritmos con el conjunto de entrenamiento se puede pasar a clasificar nuevos datos. Para ello se ha creado un script que va cargando todos los archivos LAS presentes en una carpeta y aplica los algoritmos elegidos a cada bloque generando a la salida en una nueva carpeta un archivo LAS con el nombre original del archivo más una terminación en función del algoritmo que ha generado cada archivo. Estos ficheros presentarán una clasificación de todos los puntos en dos categorías o clases; valor 0, no cable y valor 1, cable.

Debemos comentar en esta parte del trabajo que podrían incluirse en la carpeta de entrada todos los bloques LIDAR del proyecto realizado en 2017 pero por motivos de coste computacional y espacio físico solo se ha realizado con algunos ficheros seleccionados del total de archivos presentes en el proyecto. Como hemos comentado anteriormente el proyecto cuenta con aproximadamente 80.000 bloques de 1km de lado cada uno con un tamaño comprendido entre 500MB y 6GB, al aplicar 6 algoritmos que han sido los elegidos por obtener mejores resultados en las áreas de entrenamiento para detectar cables en nuevas zonas se generan 6 archivos con igual tamaño que el fichero original por lo tanto al aplicar el algoritmo en un archivo de 6GB se obtienen en total 36GB adicionales de archivos.

Además del espacio necesario, la detección de líneas de alta tensión en un bloque entero de 1km lleva un tiempo de entre 20 minutos y 1 hora en función del algoritmo utilizado y el tamaño del bloque seleccionado por lo tanto el tiempo necesario para la realización de la clasificación de todo el proyecto sería muy grande en relación al tiempo del que se dispone para realizar el presente trabajo de investigación.

Para la realización del proceso de clasificación de nuevas zonas se han seleccionado 5 bloques uno de cada parte del territorio del proyecto con una casuística diferente, bloques de bosque, bloques urbanos y bloques de cultivo. Además, se han seleccionado bloques con alta densidad de puntos (30 puntos/m2 de media) y bloques con una densidad de puntos menor (10 puntos/m2 de media). Los resultados de esta clasificación pueden verse en el apartado de resultados del presente trabajo.

4- RESULTADOS

4.1 Introducción

En este capítulo del trabajo se mostrarán y analizarán los resultados de cada algoritmo de aprendizaje por separado. Para ello se calcularán las medidas de rendimiento del conjunto de test de las áreas de aprendizaje vistas en el punto anterior para cada algoritmo. Tras ello, se discutirán y analizarán los resultados obtenidos y se decidirá cuál es el algoritmo que mejor funciona para la clasificación supervisada de cables en datos LiDAR. Finalmente, se mostrarán los resultados obtenidos en áreas nuevas con las que nunca han entrenado los algoritmos para observar los resultados obtenidos en la clasificación de los nuevos datos LiDAR.

4.2 Resultados de cada algoritmo de clasificación en las áreas de entrenamiento

En esta primera parte del capítulo resultados se muestra la matriz de confusión de los algoritmos utilizados con los datos del conjunto de test anteriormente creado. Parece interesante incluir tanto las matrices de confusión obtenidas con todos los algoritmos sin balancear el conjunto de entrenamiento y las matrices obtenidas tras aplicar el balanceo de los datos.

La matriz de confusión de cada algoritmo podemos verla a continuación.

DATOS NO BALANCEADOS

KNN		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	117.891	30.091
	NO CABLE	117.836	866.113

Random Forest		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	42.282	22.200
	NO CABLE	21.856	1.045.593

Árbol decisión – C 4.5		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	42.679	25.303
	NO CABLE	28.818	1.035.131

Bagging		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	39.282	23.203
	NO CABLE	15.131	1.054.315

Árbol decisión - CART		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	37.676	20.306
	NO CABLE	29.521	1.044.428

Adaboost		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	39.349	20.065
	NO CABLE	12.661	1.059.856

Tabla 13: Matriz de confusión de cada algoritmo con las áreas de entrenamiento no balanceadas. Fuente: Elaboración propia.

En las matrices de confusión de los datos no balanceados se observa que se presenta una gran desigualdad entre los puntos clasificados como no cable y cable. Tras observar estos datos se decide balancear el conjunto de entrenamiento. Los resultados obtenidos tras balancear el conjunto de entrenamiento se pueden ver en las siguientes matrices de confusión (tabla 14):

DATOS BALANCEADOS

KNN		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	17.891	5.091
	NO CABLE	42.836	1.066.113

Random Forest		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	19.282	4.700
	NO CABLE	3.356	1.104.593

Árbol decisión – C 4.5		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	17.679	5.303
	NO CABLE	8.818	1.100.131

Bagging		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	18.782	4.200
	NO CABLE	4.400	1.104.549

Árbol decisión - CART		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	17.676	5.306
	NO CABLE	9.521	1.099.428

Adaboost		Clasificación como	
		CABLE	NO CABLE
Clase real	CABLE	25.349	4.633
	NO CABLE	5.229	1.096.720

Tabla 14: Matriz de confusión de cada algoritmo con las áreas de entrenamiento balanceadas. Fuente: Elaboración propia.

En las matrices obtenidas tras el balanceo de los datos se observa que las clases están más equilibradas. En el análisis de los resultados se discutirá si esto mejora la clasificación.

MÉTRICAS DE RENDIMIENTO

	DATOS NO BALANCEADOS				DATOS BALANCEADOS			
	Accuracy	Recall	Precisión	Especificidad	Accuracy	Recall	Precisión	Especificidad
KNN	86.93%	79.67%	50.01%	88.02%	95.77%	77.85%	29.46%	96.14%
Árbol decisón - C 4.5	95.22%	62.78%	59.69%	97.29%	98.75%	76.93%	66.72%	99.20%
Árbol decisión - CART	95.60%	64.98%	56.07%	97.25%	98.69%	76.91%	64.99%	99.14%
Random Forest	99.35%	65.57%	65.92%	97.95%	99.29%	80.40%	85.18%	99.70%
Bagging	96.61%	62.87%	72.19%	98.59%	99.24%	81.72%	81.02%	99.60%
Adaboost	97.11%	66.23%	75.66%	98.82%	99.13%	84.55%	82.90%	99.53%

Tabla 15: Medidas de rendimiento calculadas para cada algoritmo. Fuente: Elaboración propia.

En la tabla anterior podemos ver las métricas de rendimiento anteriormente explicadas para los datos balanceados y no balanceados. En la parte de análisis de resultados se discutirán estos resultados.

Debido a la dificultad de interpretación de todos los datos anteriores se ha decidido buscar una medida de rendimiento que muestre mejor la idoneidad de cada algoritmo de manera más visual. Para ello se ha recurrido a calcular la media geométrica. Este parámetro puede describirse como una métrica que permite obtener un balance entre los porcentajes de ejemplos bien clasificados de las dos clases. Los resultados podemos verlos en la siguiente tabla.

	Media Geométrica	
	NO BALANCEADOS	BALANCEADOS
KNN	83,74 %	86,51 %
Árbol decisión - C 4.5	78,15 %	87,36 %
Árbol decisión - CART	79,49 %	87,32 %
Random Forest	80,14 %	89,53 %
Bagging	78,73 %	90,22 %
Adaboost	80,90 %	91,73 %

Tabla 16: Media geométrica de cada algoritmo de clasificación. Fuente: Elaboración propia.

4.3 Análisis de los resultados de las áreas de entrenamiento.

Observando las medidas de rendimiento de cada algoritmo se concluye que existen grandes diferencias entre ellos. Si nos centramos en los resultados correspondientes al Accuraccy (puntos de cable y no cable bien clasificados respecto al total de puntos) vemos que gran parte de los algoritmos presentan unos valores por encima del 99% pero como hemos comentado anteriormente estos valores no son representativos ya que existe una gran diferencia de número de puntos de no cable y de cable, por lo tanto, debemos calcular medidas de rendimiento que representen mejor la bondad de cada algoritmo.

Una medida más realista sobre lo bien que clasifica los puntos de cable cada algoritmo sería la Precisión, como sabemos este valor indica la proporción de ejemplos clasificados en la clase positiva que son realmente de la clase positiva. Estos resultados podemos verlos en la tercera columna de cada tipo de datos (balanceados y no balanceados) y se puede ver que existen grandes diferencias entre los diferentes algoritmos.

Destacar que los algoritmos obtienen mejores resultados con los datos balanceados. Como hemos podido leer en la bibliografía los algoritmos que se enfrentan a conjuntos de datos no balanceados tienden a predecir más valores en la clase mayoritaria ya que estadísticamente cometerán menos errores (Sanz-Delgado, *2018*). Por lo tanto, es lógico que los algoritmos cometan más errores en los datos no balanceados.

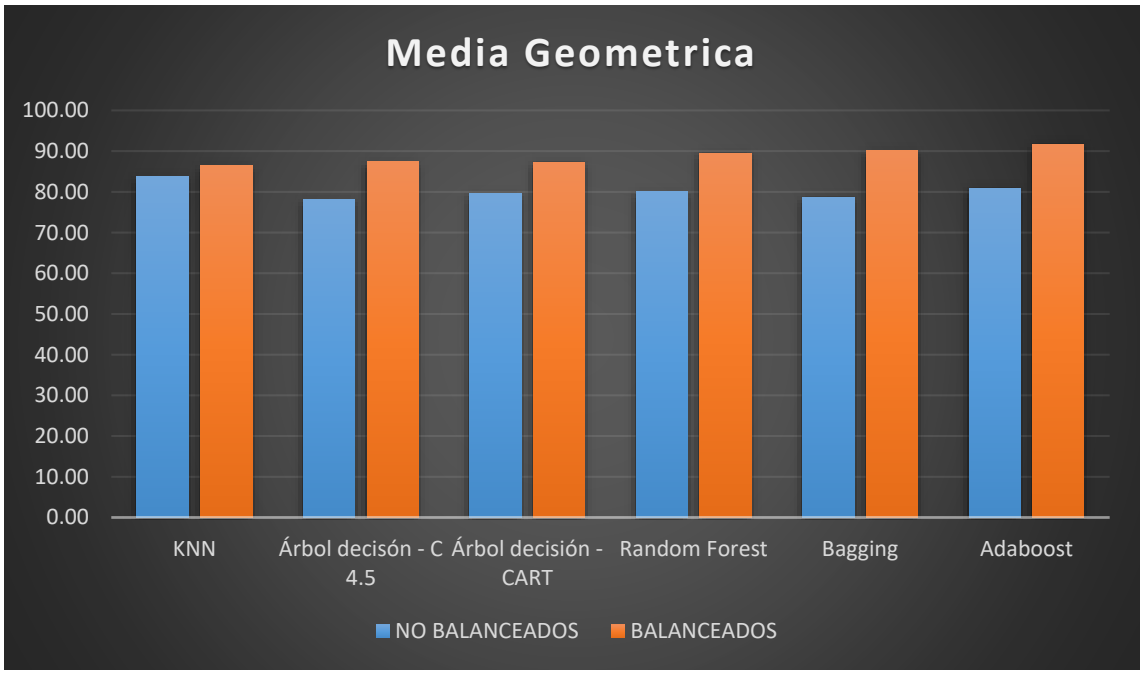


Figura 46: Grafica con las medias geométricas de cada algoritmo de clasificación. Fuente: Elaboración propia.

Para elegir el mejor algoritmo debemos fijarnos en los resultados de la media geométrica de la última tabla ya que el mejor modelo debería ser el que clasifique un mayor número de puntos pertenecientes a cables como cables, pero sin clasificar muchos puntos que no pertenezcan a cables. Para mostrar mejor estos resultados se ha creado un gráfico con los datos de *Media geométrica* tanto de los datos balanceados como de los no balanceados.

Como se puede observar en la gráfica los algoritmos basados en arboles de decisión presentan mejores resultados que los algoritmos basados en vecindad. También podemos ver que los modelos que utilizan más de un árbol de decisión (*Random Forest, Adaboost, Bagging*) obtienen mejores resultados que los algoritmos basados en un solo árbol de decisión.

4.4 Resultados y análisis de la clasificación de nuevos datos.

Para determinar la clasificación de nuevos datos con los que no han trabajado los algoritmos debemos visualizar el fichero LAS una vez que lo ha clasificado cada algoritmo y observar si existen zonas de cables bien clasificadas. Dado que no disponemos de valores de clasificación en estos ficheros no podremos obtener medidas de rendimiento para evaluar cada algoritmo, únicamente evaluaremos cada algoritmo si los cables presentes en los ficheros nuevos se encuentran bien clasificados visualmente.

La siguiente imagen es un recorte de un bloque LAS de 1km de lado en que existen dos cables. En este caso solo nos debemos fijar en el cable relativo a la línea de alta tensión ya que los algoritmos solo han entrenado con puntos pertenecientes a líneas de alta tensión. En la imagen 47 esta se representa por un trazo blanco más ancho. Para observar mejor la línea de alta tensión presente en ella vamos a visualizarla tanto en planta como en 3D. En la imagen se ha eliminado el ruido de la visualización para poder observar el cable. A continuación, veremos cómo ha clasificado cada algoritmo esta zona.

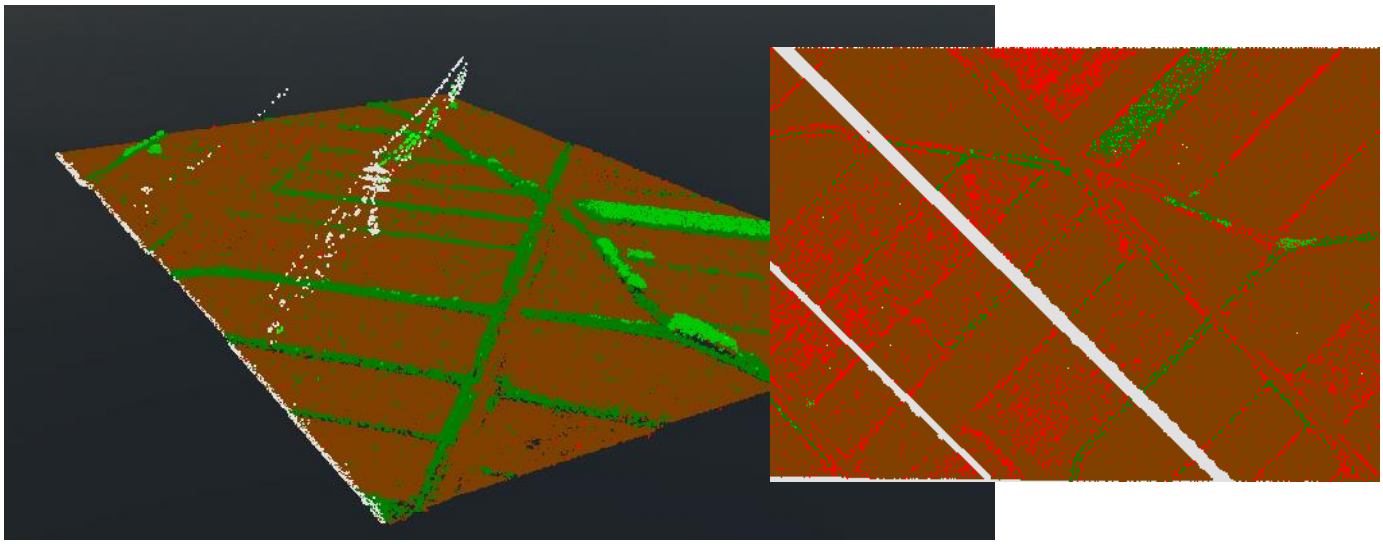
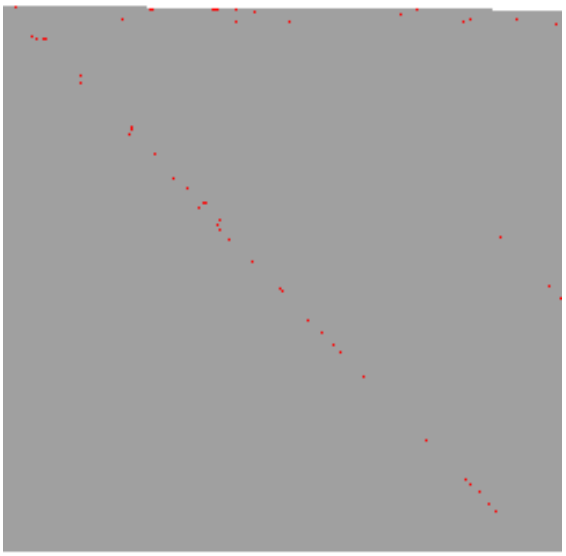


Figura 47: Recorte de bloque LAS utilizado para la valoración de la clasificación. Fuente: Elaboración propia.

Los resultados de la clasificación por cada algoritmo de la zona anterior se muestra en las siguientes imágenes. (Figuras 48,49,50).

1-Adaboost



2-Árbol decisión- C4.5

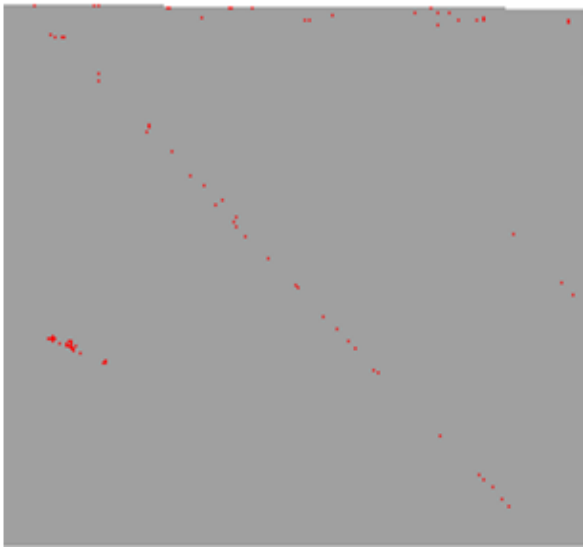
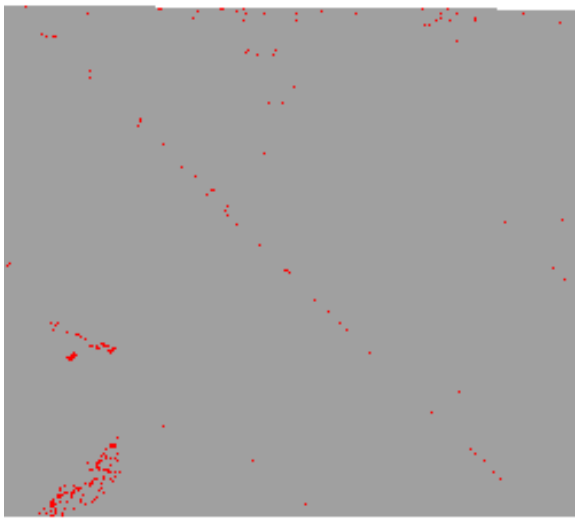


Figura 48: Resultado de la clasificación de la imagen anterior. Fuente: Elaboración propia.

3- Árbol Decisión- GINI



4- Bagging

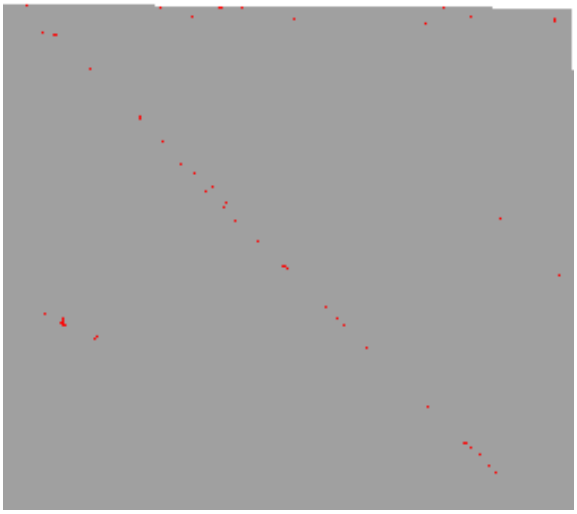
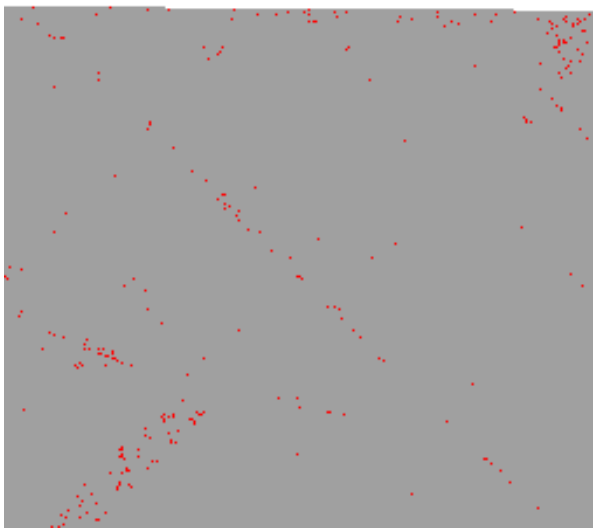


Figura 49: Resultado de la clasificación de la imagen anterior. Fuente: Elaboración propia.

5- KNN



6-Random Forest

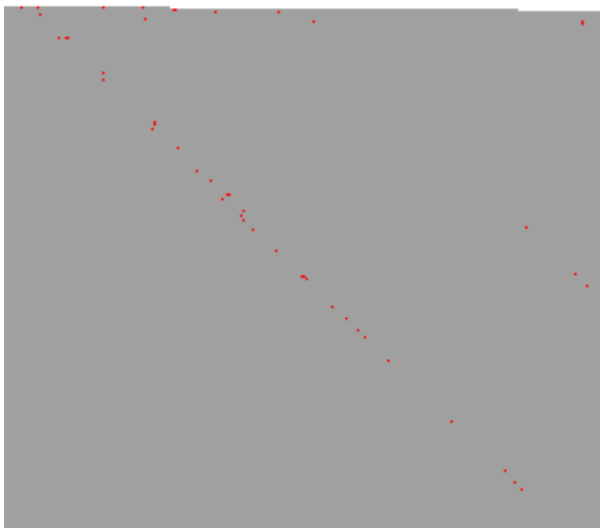


Figura 50: Resultado de la clasificación de la imagen anterior. Fuente: Elaboración propia.

Como se puede observar en las imágenes anteriores todos los algoritmos han conseguido encontrar los cables de la línea de alta tensión presente en la imagen. Aunque todos los algoritmos han conseguido encontrar los cables debemos comentar que la diferencia entre unos y otros es que algunos algoritmos como *Random Forest y Adaboost* han conseguido clasificar bien los cables y además han clasificado bien los puntos de no cables, por ello serían los mejores algoritmos para el trabajo que deseamos realizar.

Otros algoritmos utilizados también han conseguido clasificar los puntos de los cables, aunque también han incluido multitud de puntos pertenecientes a no cable y los han clasificado como cables. Esta sería la cualidad que diferencia estos algoritmos ya que se cree que es mejor que clasifique menos puntos de cada cable, pero estos se encuentren bien clasificados.

En la imagen siguiente se puede observar uno de las zonas clasificadas por los algoritmos en 3 dimensiones para observar los puntos clasificados como cable. Esta imagen concretamente corresponde al modelo predicho por el algoritmo *Random Forest*.

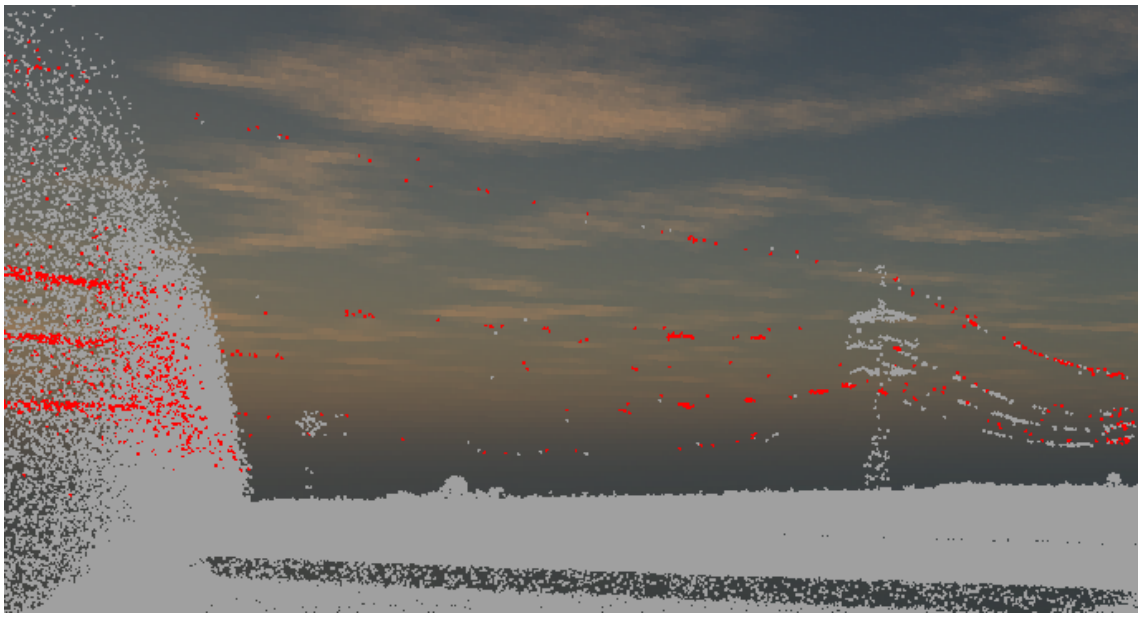


Figura 51: Resultado de la clasificación de la imagen anterior en 3D. Fuente: Elaboración propia.

5-CONCLUSIONES

5.1 Conclusiones de la detección.

En esta parte del trabajo se expondrán las conclusiones a las que se ha llegado tras observar los resultados obtenidos en el presente trabajo de investigación. Como se ha podido observar en las imágenes anteriores y tras analizar el resto de datos se ha conseguido la detección de cables en los datos LIDAR del proyecto utilizado mediante algoritmos de clasificación supervisada únicamente con los datos presentes en el archivo LAS de cada bloque. Quedaría comprobar si estos algoritmos son extrapolables a otro tipo de datos LIDAR diferentes a los utilizados en el presente trabajo de investigación.

La primera conclusión y más importante a la que se ha llegado es que es necesario un balanceo de los datos de entrenamiento para entrenar a los algoritmos de clasificación supervisada ya que como hemos observado en los datos obtenidos sin balancear el conjunto de entrenamiento los resultados son sustancialmente peores y en la detección de cables en nuevas zonas presentan múltiples errores clasificando generalmente los puntos como no cable. Aunque obtienen rendimientos totales altos éstos no son realistas debido al escaso balanceamiento de los datos y al gran número de aciertos al clasificar los puntos como no cable.

Otra conclusión importante a la que se ha llegado ha sido la dificultad de trabajar con estos datos LIDAR obtenidos con sensores SPL debido a la gran cantidad de ruido que presentan. Como se ha expuesto anteriormente no se ha podido realizar le creación de variables auxiliares en la clasificación de cables a partir del propio fichero LAS mediante triangulación TIN ni creación de MDE como se ha realizado en otros trabajos consultados, debido a la gran cantidad de ruido que presentan estos datos y a la no existencia de una clasificación correcta de estos datos, antes de empezar el trabajo para poder crear un MDE correcto con el que seguir la creación de variables auxiliares. Tampoco se ha podido realizar un raster de cada bloque para buscar elementos lineales como en otros trabajos debido a la alta cantidad de puntos correspondiente a ruido sobre el suelo y los cables

Además, puede que exista la posibilidad de que las variables con las que los algoritmos han trabajado no sean suficientes y que además estas no sean del todo correctas. Aunque se han encontrado cables en algunos de los nuevos bloques clasificados por los algoritmos también han existido otros bloques con exceso de ruido en los que los algoritmos no han funcionado tan bien. Creemos que existen datos como el valor de intensidad que presenta valores muy diferentes para una misma superficie y aunque se supone que se encuentra normalizada para todos los dataos se han encontrado incoherencias en ellos. Las variables R, G, B que hacen referencia al valor de cada color en cada punto han sido tomadas de una imagen capturada en el mismo momento de la toma de datos LIDAR, pero como se ha comprobado posteriormente puede que esta información este desplazado en algunos puntos por lo que puede que las variables de color tampoco sean del todo correctas en algún punto lo que hace confundir a los algoritmos.

Por otro lado, toda la bibliografía que se ha consultado realiza la clasificación en datos LIDAR multifotónicos con unas características muy diferentes a las de los datos de LIDAR SPL utilizados en este trabajo. La mayor densidad de puntos debería ser una ayuda para la detección de cables, pero no así la gran cantidad de ruido que presentan estos datos que dificulta en gran medida la realización del trabajo. Creemos que al ser este sensor un sistema novedoso se necesita más tiempo y estudios para conseguir unos mejores resultados con este tipo de datos.

Por ultimo también existe la posibilidad de que se haya realizado un sobre entrenamiento o sobreajuste de los modelos utilizados con las áreas de entrenamiento y por lo tanto no clasifiquen tan bien los nuevos datos como cabría esperar ya que tras observar las medidas de rendimiento de los algoritmos con las áreas de entrenamiento se esperaban unos resultados mejores en la detección de cables en nuevos datos.

5.2 Futuras líneas de investigación.

Como futuras líneas de investigación tras la realización de este trabajo y siguiendo la misma línea a la que se ha planteado estaría la creación de nuevas variables dependientes de las anteriores para intentar mejorar la clasificación de cables. Se ha pensado por ejemplo en la creación de variables en cada punto dependientes de los puntos que le rodean realizando una media o un valor ponderado en función de la distancia de cada variable obtenida a partir de un número de puntos vecinos concreto e intentar así mejorar la clasificación. También se cree que mejoraría la clasificación la creación de una variable que mostraría la altura de cada punto respecto al suelo y no la coordenada Z total utilizada en este trabajo.

Por otro lado, se cree que con otro tipo de algoritmos basados en la geometría de cada punto respecto a los demás podría realizar una mejor clasificación. Se ha pensado en la creación de una esfera alrededor de cada punto e intentar trazar líneas de lado a lado de la esfera pasando por el punto central y por dos puntos de dicha esfera, si esto se consigue el siguiente paso es intentar seguir el modelo con los puntos que formen parte de la línea y de este modo seguir buscando la línea que pertenezca al cable.

6-BIBLIOGRAFIA

Albacete, A. S. J. (2011). Facultad de Geografía e Historia Procesamiento de datos LiDAR con ArcGIS Desktop 10 Autor : Antonio San José Albacete Director : Luís Garmendia Salvador Codirector : Francisco Mauro Gutiérrez Madrid 2011, (August).

Alexander, C., Tansey, K., Kaduk, J., Holland, D., & Tate, N. J. (2011). An approach to classification of airborne laser scanning point cloud data in an urban environment. *International Journal of Remote Sensing*, *32*(24), 9151–9169. https://doi.org/10.1080/01431161.2010.550645

ASPRS. (2013). LAS Specification Version 1.4-R13. *The American Society for Photogrammetry & Remote Sensing*, *July*(November 2011), 1–28. https://doi.org/faf

Axelsson, P. (1999). Processing of laser scanner data—algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, *54*, 138–147. https://doi.org/10.1016/S0924-2716(99)00008-8

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Full-Text, 1–33. https://doi.org/10.1017/CBO9781107415324.004

Chehata, N., Guo, L., & Mallet, C. (2009). Airborne Lidar feature Selection for urban classification using Random Forests. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XXXVIII*(Part 3 / W8), 207–212.

Clode, S., & Rottensteiner, F. (2005). Classification of trees and powerlines from medium resolution airborne laserscanner data in urban environments. *… of the APRS Workshop on Digital …*, (February), 191–196. Retrieved from http://archive.itee.uq.edu.au/~aprs/wdic2005/fullproceedings.pdf#page=36

Collin, A., Archambault, P., & Long, B. (2011). Predicting species diversity of benthic communities within turbid nearshore using full-waveform bathymetric LiDAR and machine learners. *PLoS ONE*, *6*(6). https://doi.org/10.1371/journal.pone.0021265

Dawe, H. G., & Engineer, C. P. (1947). LARGE SCALE HIGH PRECISION MAPPING BY PHOTOGRAMMETRIC METHODS, *480*, 142–152.

Degnan, J. J., Field, C., Machan, R., Leventhal, E., Lawrence, D., Zheng, Y., … Howell, S. (2007). Recent Advances in Photon-Counting , 3D Imaging Lidars, (November), 8–13.

Developers, S. (2017). Scikit-Learn User Guide. Retrieved from http://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf

Fagua, J., Campo, A., & Posada, E. (2011). Desarrollo de dos metodologías para la generación de modelos digitales de terreno (MDT) y superficie (MDS) empleando datos LiDAR y programas de licencia. *Análisis Geográfico*, *49*(May), 83–95.

Garcia-Gutierreza, J., Concalves-Seco, L., & Riquelme-Santos, J. (2009). Decision trees on LiDAR to classify land uses and covers. *Iaprs*, *XXXVIII*, 323–328. Retrieved from http://www.isprs.org/proceedings/XXXVIII/3-W8/papers/323_laserscanning09.pdf

Gwenzi, D., & Lefsky, M. A. (2014). Prospects of photon counting lidar for savanna ecosystem structural studies. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *40*(1), 141–147.

https://doi.org/10.5194/isprsarchives-XL-1-141-2014

Han, J., & Kamber, M. (2000). Data Mining: Concepts and Techniques. *Data Mining:Concepts and Techniques*, 3–26. https://doi.org/10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C

Harding, D. J., Dabney, P. W., & Valett, S. (2011). Polarimetric, two-color, photon-counting laser altimeter measurements of forest canopy structure. *International Symposium on Lidar and Radar Mapping Technologies*, 828629. https://doi.org/10.1117/12.913960.Reflected

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844. https://doi.org/10.1109/34.709601

Holm, S. (1979). Board of the Foundation of the Scandinavian Journal of Statistics A Simple Sequentially Rejective Multiple Test Procedure A Simple Sequentially Rejective Multiple Test Procedure. *Source: Scandinavian Journal of Statistics Scand J Statist*, *6*(6), 65–70. https://doi.org/10.2307/4615733

Jes, J. M. De. (2016). propagation en problemas de desbalance de clases : Un estudio empírico sobre la clasificación en imágenes de perc ... Técnicas de muestreo para mejorar el rendimiento del algoritmo back- propagation en problemas de desbalance de clases : Un estudio empíri, (March).

Jwa, Y., & Sohn, G. (2012). A Piecewise Catenary Curve Model Growing for 3D Power Line Reconstruction. *Photogrammetric Engineering & Remote Sensing*, *78*(12), 1227–1240. https://doi.org/10.14358/PERS.78.11.1227

Jwa, Y., Sohn, G., & Kim, H. B. (2009). AUTOMATIC 3D POWERLINE RECONSTRUCTION USING AIRBORNE LiDAR DATA. *Iaprs*, *XXXVIII*(2004), 105–110.

Kim, H. B., & Sohn, G. (2011). Random Forests Based Multiple Classifier System for Power-Line Scene Classification. *International Archives of the Photogrammetry*, *XXXVIII*(August), 29–31.

Landa, A. F. A. A. F., Rodríguez, F., López, D., Olabarria, J. R. G., Yudego, B. M., Lasala, D., … Molina, A. G. (2013). Los sensores aerotransportados LiDAR y multiespectrales en la descripción y cuantificación de los recursos forestales. *Revista Montes.*, (112), 31–36. Retrieved from http://dialnet.unirioja.es/servlet/articulo?codigo=4194545

Li, Z., Liu, Y., Walker, R., Hayward, R., & Zhang, J. (2010). Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved Hough transform. *Machine Vision and Applications*, *21*(5), 677–686. https://doi.org/10.1007/s00138-009-0206-y

Marsh, B. (2016). Multivariate Analysis of the Vector Boson Fusion Higgs Boson, (August).

Martínez Blanco, M. (2016). Evaluación y propuesta de metodologías de clasificación a partir del procesado combinado de datos LiDAR e imágenes aéreas georreferenciadas, 225. Retrieved from http://repositorio.unican.es/xmlui/bitstream/handle/10902/8339/TesisMPMB.pdf?sequence=1&isAllowed=y

McLaughlin, R. A. (2006). Extracting transmission lines from airborne LIDAR data. *IEEE Geoscience and Remote Sensing Letters*, *3*(2), 222–226. https://doi.org/10.1109/LGRS.2005.863390

Melzer, T., & Briese, C. (2004). Extraction and Modeling of Power Lines from ALS Point Clouds. *28th Workshop of the Austrian Association for Pattern Recognition (ÖAGM)*, 8.

Mozo, M. C., & Alconada, M. I. M. (n.d.). GT-14.

Mozo, M. C., & Alconada, M. I. M. (2014). Control de calidad del vuelo Lidar utilizado para la modelizaci{{}{ó}{}}n 3D de las fallas de Alhama (Murcia) y Carboneras (Almer{{}{í}{}}a). Retrieved from http://oa.upm.es/33673/

Niemeyer, J., Rottensteiner, F., & Soergel, U. (2013). Classification of urban LiDAR data using conditional random field and random forests. *Proceedings of the JURSE 2013*, *856*(1), 139–142. https://doi.org/10.1109/JURSE.2013.6550685

Perdomo, J. G. (2007). Minería de Datos II, 284.

Photon, S., Brings, L., Pulse, H., For, R., & Lidar, A. (n.d.). SINGLE PHOTON LIDAR BRINGS HIGHER PULSE RATES FOR AIRBORNE LIDAR The Evolution of Lidar, (Figure 1).

Robles Porcada, V. (2003). Redes Bayesianas. Aplicación en Biología Computacional, 210.

Rokach, L., & Maimon, O. (2008). Data Mining With Decision Trees: Theory and Applications. *Series in Machine Perception and Artificial Intelligence, World Scientific*, *1*, 78–88.

Romero, R. M., Merino, S., Miguel, D., Magdaleno, F., Ingeniería, Á. De, Cedex, A., … Xii, A. (2009). (Lidar) En Hidrología Forestal Y En La Gestión De Ecosistemas Fluviales, *27*, 23–27.

Sohn, G., Jwa, Y., & Kim, H. B. (2012). Automatic Powerline Scene Classification and Reconstruction Using Airborne Lidar Data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *I*(September), 167–172. https://doi.org/10.5194/isprsannals-I-3-167-2012

Soininen, A. (2016). TerraScan User's Guide ---, 2–592.

Swatantran, A., Tang, H., Barrett, T., Decola, P., & Dubayah, R. (2016). Rapid, high-resolution forest structure and terrain mapping over large areas using single photon lidar. *Scientific Reports*, *6*(May), 1–12. https://doi.org/10.1038/srep28277

Visa, S., & Ralescu, A. (2003). Learning imbalanced and overlapping classes using fuzzy sets. *Workshop on Learning from Imbalanced Datasets II (ICML '03)*, (0), 91–104.

Vosselman, G., & Maas, H.-G. (2010). Airborne and Terrestrial Laser Scanning, 318. Retrieved from http://site.ebrary.com/lib/aghkrakow/detail.action?docID=10698310

Wang, Y., Chen, Q., Liu, L., Zheng, D., Li, C., & Li, K. (2017). Supervised classification of power lines from airborne LiDAR data in Urban Areas. *Remote Sensing*, *9*(8), 8–11. https://doi.org/10.3390/rs9080771

Witten, I. H., & Frank, E. (2005). Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations, (January 2006), 55860–552. https://doi.org/10.1186/1475-925X-5-51

Wulder, M., White, J. C., Wulder, M. A., Bater, C. W., Coops, N. C., Hilker, T., & White, J. C. (2008). The role of LiDAR in sustainable forest management The role of LiDAR in sustainable forest management, *84*(September 2016), 807–826. https://doi.org/10.5558/tfc84807-6

Yang, B., Wei, Z., Li, Q., & Li, J. (2012). Automated extraction of street-scene objects from mobile lidar point clouds. *International Journal of Remote Sensing*, *33*(18), 5839–5861.

https://doi.org/10.1080/01431161.2012.674229

Zhang, J., Lin, X., & Ning, X. (2013). SVM-Based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sensing*, *5*(8), 3749–3775. https://doi.org/10.3390/rs5083749

Zhu, L., & Hyyppa, J. (2014). The use of airborne and mobile laser scanning for modeling railway environments in 3D. *Remote Sensing*, *6*(4), 3075–3100. https://doi.org/10.3390/rs6043075

Zhu, L., & Hyyppä, J. (2014). Fully-automated power line extraction from airborne laser scanning point clouds in forest areas. *Remote Sensing*, *6*(11), 11267–11282. https://doi.org/10.3390/rs61111267